New Directions in Approximation Algorithms and Hardness of Approximation

Yuan Zhou

CMU-CS-14-125
August 2014

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Venkatesan Guruswami, Co-chair
Ryan O'Donnell, Co-chair
Anupam Gupta
R. Ravi
Sanjeev Arora (Princeton University)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored by the National Science Foundation under grant numbers CCF-0915893, CCR-0953155, CCF-1115525; US Israel Binational Science Foundation under grant number 2008293; Simons Foundation under grant number 252545; and generous support from the Microsoft Corporation.

Keywords: Approximation Algorithms, Hardness of Approximation, Combinatorial Optimization, Convex Relaxation Hierarchies, Linear Programming, Semidefinite Programming

DEDICATION

To my parents.

Abstract

Combinatorial optimization encompasses a wide range of important computational tasks such as UNIFORMSPARSESTCUT (also known as NORMALIZEDCUT), MAXCUT, TRAVELINGSALESMANPROBLEM, and VERTEXCOVER. Most combinatorial optimization problems are NP-hard to be solved optimally. On one hand, a natural way to cope with this computational intractability is via designing *approximation algorithms* to efficiently approximate the optimal solutions with provable guarantees. On the other hand, given an NP-hard optimization problem, we are also interested in the best possible approximation guarantee that any polynomial-time algorithm could achieve, i.e. the *hardness of approximation* of the problem. Both approximation algorithms and hardness of approximation results contribute to understanding the *approximability* of combinatorial optimization problems.

In the last two decades, the research frontier of approximation algorithm design has been greatly advanced thanks to the convex optimization techniques such as linear and semidefinite programming. However, the exact approximability for many problems remains mysterious but some common barriers for progress revolving around a problem called Unique Games has been identified. The limitations of convex relaxation techniques answer the question that what is the best possible approximation guarantee to be achieved by the state-of-the-art algorithmic design tools, and shed light on the real approximability. Therefore, the study of the power of convex relaxations becomes a valuable new research direction to get around the current barrier on hardness proofs.

In this thesis, using constraint satisfaction problems, assignment problems, graph partitioning problems (BALANCEDSEPARATOR, UNIFORMSPARSESTCUT, DENSEkSUBGRAPH), and graph isomorphism as examples, we explore both the effectiveness and limitations of the most powerful convex relaxation techniques – convex relaxation *hierarchies*. We also use a proof complexity view of the convex relaxation hierarchies to analyze their performance on constraint satisfaction problems, and show that the so-called Parrilo–Lasserre semidefinite programming relaxation hierarchy succeeds on all hard instances constructed in literature for UNIQUEGAMES, MAXCUT, and BALANCEDSEPARATOR.

This thesis also contains a collection of approximation algorithms for almost satisfiable constraint satisfaction problems and MAXBISECTION, detection of almost isomorphic trees, and estimation of the $2 \to 4$ operator norm of random linear operators. There are also a few (conditional) hardness of approximation results for almost satisfiable linear systems over integers, almost satisfiable MAXHORN3-SAT, and detection of almost isomorphic graphs.

Acknowledgments

It would not have been possible to complete this doctoral thesis without the help and support of many people around me, to only some of whom it is possible to give particular mention here.

Above all, I owe my deepest gratitude to my Ph.D. advisors Venkat Guruswami and Ryan O'Donnell. When still in college, I read and was intrigued by a set of beautifully scribed online lecture notes on the PCP theorem, approximation algorithms, and hardness of approximation. A year later, I was fortunate enough to become a student of both of the lecturers in that course. In fact, Venkat and Ryan are the best advisors I could have asked for. I was always amazed by their brilliance, technical mastery, and wide knowledge on almost everything. Venkat could always patiently go through every ridiculous ideas of mine; and Ryan never failed to figure a new direction after we hitting thousands of dead-ends. Over the years, I enjoyed all the discussions through which I learned a lot from their techniques and research styles. For all these and the freedom in choosing research topics, the career advices, and a lot more left out – Thank you Venkat and Ryan!

I am very grateful to my thesis committee: Venkatesan Guruswami, Ryan O'Donnell, Sanjeev Arora, Anupam Gupta, and R. Ravi. Thank you for taking time from your busy schedules to make my thesis defense.

Many thanks to each of my paper co-authors through my Ph.D. : Boaz Barak, Aditya Bhaskara, Fernando Brandão, Moses Charikar, Xi Chen, Julia Chuzhoy, Parikshit Gopalan, Venkatesan Guruswami, Aram Harrow, Zhiyi Huang, Manuel Kauers, Jonathan Kelner, Gabor Kun, Jian Li, Yury Makarychev, Raghu Meka, Ryan O'Donnell, Prasad Raghavendra, Omer Reingold, Ali Kemal Sinop, David Steurer, Suguru Tamaki, Li-Yang Tan, Madhur Tulsiani, Salil Vadhan, Aravindan Vijayaraghavan, Lei Wang, John Wright, Chenggang Wu, Yi Wu, Yu Wu, Yuichi Yoshida, and Jiawei Zhang. It was great joy working with you; and this thesis would not be possible without your collaborations.

I had the rare opportunity to spend each of the four summers during graduate school visiting a number of places. When still getting familiar with theoretical research in my first year, I had the pleasure of interning with Julia Chuzhoy and Yury Makarychev at Toyota Technological Institute at Chicago, as well as visiting Pinyan Lu at Microsoft Research Asia. In the following summers, I enjoyed collaborating with Boaz Barak at Microsoft Research New England, Parikshit Gopalan at Microsoft Research Silicon Valley, and Yury Makarychev and Madhur Tulsiani at Toyota Technological Institute at Chicago. During winter breaks, I also enjoyed a couple of visits to Shengyu Zhang at the Chinese University at Hong Kong, Yitong Yin at Nanjing University, and Xiaoming Sun at the Chinese Academy of Sciences. I am greatly indebted to all these institutions I have visited and the numerous people who hosted me. I would also like to thank the support from the Simons Graduate Fellowship in Theoretical Computer Science during the last two years.

I wish to thank my high school teacher Tao Jiang for introducing me to the world of algorithmic programming contest. I am also greatly indebted to Professor Andrew Yao and Elad Verbin at Tsinghua University and Wei Chen at Microsoft Research Asia who further guided my interest to theoretical computer science.

Many thanks go to the theory group faculty members at CMU from whom I learned a lot: Avrim Blum, Manuel Blum, Alan Frieze, Anupam Gupta, Gary Miller, and Danny Sleator.

Also, special thanks to the friends I met in Pittsburgh and during my summer internships for making my Ph.D. life colorful: Parinya Chalermsook, Zhuo Chen, Ricky Chis, Bin Fu, Edinah Gnang, Fan Guo, Favonia Hou, Jason Lee, Lei Li, Nan Li, Jian Peng, Richard Peng, Julian Shun, Xiaorui Sun, Carol Wang, John Wright, Chenye Wu, Yi Wu, Guangyu Xia, Guang Xiang, Lin Xiao, Ning Xie, Jia Xu, Kathy Zhang, Zeyuan Zhu, Mengjie Zou.

Finally, it would be difficult for me to express the debt of gratitude that I owe my parents Qihua Zhou and Hangli Wang, for their love and constant support.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Many important computational tasks can be modeled as combinatorial optimization problems, where the goal is to optimize a certain objective function on discrete variables subject to some constraints. To give the readers a flavor of the optimization problems studied in this thesis, we list a few examples as follows.

Problem 1 (the $\frac{1}{3}$-BALANCEDSEPARATOR problem)**.** *Given an undirected graph, partition the set of vertices into two parts so that each part contains at least $\frac{1}{3}$ of the total vertices and the number of edges across the partition is* minimized*.*

There are many similar problems as BALANCEDSEPARATOR (such as MAXCUT, UNIFORMSPARSESTCUT), which consist of the class of graph partitioning problems, and arise in many settings such as clustering, divide and conquer algorithms, VLSI layout, etc. There are also many other well-studied optimization problems, including the following examples.

Problem 2 (the TRAVELINGSALESMANPROBLEM problem)**.** *Given a list of cities and the distances between each pair of cities, find out the* shortest *possible route (i.e. the one with* minimum *total distance) that visits each city exactly once and returns to the origin city.*

Problem 3 (the VERTEXCOVER problem)**.** *Given an undirected graph, find out the* smallest *set of vertices (i.e. the one with* minimum *cardinality) such that each edge of the graph is incident to at least one vertex of the set.*

Problem 4 (solving overdetermined sparse linear systems)**.** *Given a system of linear equations over rational numbers, so that each equation contains at most 3 variables. If the linear system is consistent, it is easy to find a solution using Gaussian elimination. However,* *suppose the system is not completely consistent, i.e. some of the equations are erroneous. The natural optimization problem here is to find a solution that satisfies the* maximum *number of equations.*

The readers may refer to Section 2.1 where more combinatorial optimization problems (which are also the problems studied in this thesis) are defined.

For many combinatorial optimization problems, it is computationally intractable (NP-hard) to find the optimal solution. A popular and extensively studied way to deal with this intractability is via designing approximation algorithms to efficiently approximate the optimal solutions with provable guarantees. We will briefly introduce this notion in Section 1.1. We will also discuss (in Section 1.1.2) the notion of hardness of approximation, i.e. the limitation on approximation guarantee for polynomial-time algorithms. While it is desirable to design an approximation algorithm and prove hardness of approximation matching the approximation guarantee given by the algorithm, we are usually not able to achieve this goal due to the limited techniques on both algorithmic analysis and hardness proofs.

On the algorithmic side, convex relaxation hierarchies, which we will introduce in Section 1.2 and Section 1.3, are the most powerful framework for designing approximation algorithms. For many important combinatorial optimization problems, the state-of-the-art approximation algorithms only use a small portion of the power of convex relaxation hierarchies (as later discussed in the subsections). Therefore, the pursue of better approximation algorithms greatly motivates the further study of convex relaxation hierarchies.

It is also worthwhile to explore the limitation of convex relaxation hierarchies, given the lack of sharp hardness of approximation results for many important problems. As discussed in Section 1.3.1, although such type of results does not work against *all* polynomial-time algorithms, they share a glimpse into the frontier of our approximation techniques, and help us understand the complexity of approximating optimization problems where no concrete NP-hardness is known. In the light of this, showing the limitation for the hierarchies can be viewed as a way of going beyond our limited NP-hardness results.

This thesis is a collection of results regarding the approximability and inapproximability of various combinatorial optimization problems, while a good portion of it is about the effectiveness and limitations of convex relaxation hierarchies. In Section 1.4, we will briefly talk about the results in the thesis to conclude this introductory section.

1.1 The notion of approximation algorithms

As we have seen, the class of combinatorial optimization problems encompasses many important computational tasks. However, for most of these interesting optimization problems (and indeed all the example problems provided above), finding the optimal solution is unfortunately computationally intractable (NP-hard).

One way to deal with this intractability is to use efficient algorithms to find *approximately* optimal solutions with provable guarantees – we call these *approximation algorithms*. For a maximization problem, we call an algorithm *α-approximation algorithm* ($\alpha < 1$) if the output value of the algorithm is at least α times the optimal value. Similarly, for a minimization problem, an *α-approximation algorithm* ($\alpha > 1$) guarantees to output a solution with value at most α times the value of the optimal solution. Clearly, for both maximization and minimization problems, when α gets closer to 1, we get better approximation guarantee.

Sometimes we would love to talk about the approximation guarantee in a more precise manner. Let us take following three scenarios in the MAXCUT problem (the problem where we are given an undirected graph and the goal is to find a partition of the vertex set so that the fraction of edges across the partition is maximize), for example –

1. given an instance where the optimal solution cuts *all* the edges, and we would like to find a partition cutting *all* the edges (approximation ratio : 1);
2. given an instance where the optimal solution cuts $.9999$ fraction of the edges, and we would like to find a partition cutting $.99$ fraction of the edges (approximation ratio : $\sim .9901$);
3. given an instance where the optimal solution cuts $.80$ fraction of the edges, and we would like to find a partition cutting $.77$ fraction of the edges (approximation ratio : $.9625$).

The first task is obviously easy (i.e. in P) since it is exactly to compute the bipartition of a bipartite graph. For the second task, since the optimal solution cuts *almost all* the edges, the input graph looks very close to a bipartite graph. It is conceivable that some bipartite graph recognition algorithm might be able to deal with a tiny fraction of the erroneous edges and output a quite good partition meeting the criteria in the task (and indeed the Geomans-Williamson algorithm [94], which will be heavily mentioned in the rest of this thesis, works in this way). For the third task, although the least quality of approximation is demanded in terms of the ratio, the task itself is NP-hard by [219]. The hardness may be attributed to the low value of the optimal solution. These examples motivate the definition of the approximation quality according to the value of the optimal solution. For the MAXCUT problem and $0 < s \leq c \leq 1$, we say an algorithm is a (c, s)-*approximation algorithm* if the algorithm outputs a partition cutting at least s fraction of the edges whenever the optimal solution has value at least c. An α-approximation algorithm is always a $(c, \alpha c)$-approximation algorithm for all $0 < c \leq 1$. However, as we see from the examples above, for some particular c, we can get a solution with value much better than αc. Please refer to Section 2.2 for the precise definitions of the approximation notions discussed here.

1.1.1 Robust algorithms

In the exemplary MAXCUT scenarios presented above, we had the intuition that if the task of recovering the *perfect* solution (i.e. the partition that cuts all the edges) can be done in polynomial time, then finding out an *almost perfect* solution if there is one might also be easy. This intuition is correct for MAXCUT, but might not be the case for other problems. However, the idea of generalizing an algorithm which works for perfect solutions to almost perfect solution motivates the notion of *robust algorithms*.

At a high level, robust algorithms extremely well-approximate an almost perfect solution when such solutions exist. To make this description more precise, let us fix a maximization problem. Usually we can normalize the objective of the maximization problem so that it always lies between $[0, 1]$ and having objective value 1 means a perfect solution. (In the MAXCUT example, one natural objective concerns about the number of edges across the partition. However, we chose to rescale this objective and made it the *fraction* of the desired edges.) We call an algorithm for the optimization problem *robust*, if there exists a function $r : [0, 1] \to [0, 1]$ satisfying $r(\epsilon) \to 0$ as $\epsilon \to 0^+$ such that whenever the input instance has objective value $(1 - \epsilon)$, the algorithm outputs a solution with value at least $(1 - r(\epsilon))$. The readers may refer to Section 2.2.1 for more definitions and discuss on robust algorithms.

1.1.2 Hardness of approximation

To complement algorithmic results for a problem, it is also interesting to study its *hardness of approximation*, i.e. the limitation of polynomial-time algorithms in terms of the approximation guarantee they can achieve. For a maximization problem, we say it is NP-hard to α-approximate the problem if any polynomial-time algorithm with α approximation guarantee implies $\mathsf{P} = \mathsf{NP}$. We will also talk about hardness of approximation under other complexity assumptions (such as the Unique Games Conjecture which will be introduced in Section 2.3). We can define similar hardness of approximation notion for minimization problems.

Ideally, for each problem of interest, we would like to have both an approximation algorithm and hardness of approximation which implies that there is no algorithm with better approximation guarantee, therefore identifying the *approximation threshold* of the problem.

1.2 The relaxation and rounding framework for designing approximation algorithms

Convex programming relaxations and rounding schemes are a standard tool to design approximation algorithms. A vast majority of known approximation algorithms are designed using this approach with only a few exceptions (e.g. [14, 90]). A significant part of this thesis is devoted to exploring the effectiveness and limitations of convex relaxations. In this section, we provide a rudimentary introduction to this powerful framework. We suggest readers to refer to relevant chapters in [221] and surveys such as [66] for more information.

1.2.1 Convex relaxations

In combinatorial optimization problems, the solution space is discrete and usually we can encode the solutions using variables those take value either 0 or 1. After such encoding, it is often straightforward to formulate the optimization problem as an integer program. However, it is NP-hard to exactly solve the integer program (as long as the original optimization problem is NP-hard). Such computational intractability stems from the non-convexity (or the integrality) of the solution space. The idea here is to relax the integral constraints to make the program tractable.

Specifically, we relax the condition that variables take values either 0 or 1 so that variables can be real numbers or even vectors. For example, a simple relaxation would be allow variables to take any real numbers in $[0, 1]$ (instead of $\{0, 1\}$ values). In this way, if the objective function and other constraints in the integer program are linear, we relax it to a tractable linear program. Other methods can be applied to deal with the integer programs when the objective function or constraints are not linear.

Above we have just shown one simple approach of deriving linear programming relaxations, while there are also many other relaxation techniques such as semidefinite programing relaxations and relaxation hierarchies. The main idea for all these approaches is to obtain a convex optimization problem by suitable relaxations so that the new problem is computational tractable.

1.2.2 Rounding schemes

It is clear that any solution for the original optimization problem is also a feasible solution for the convex relaxation (one may think about the linear programming relaxation example to get the intuition). Therefore, if the original optimization problem is a maximization problem, the optimum of the convex relaxation is always at least the optimum of the original problem. Fix a convex relaxation $\mathcal{R}$ and a specific problem instance $\mathcal{I}$, let $\mathsf{opt}(\mathcal{I})$ be the value of the optimal solution to $\mathcal{I}$; let $\mathsf{opt}_{\mathcal{R}}(\mathcal{I})$ be the value of the optimal solution to the convex relaxation of $\mathcal{I}$. We have just derived that $\mathsf{opt}(\mathcal{I}) \leq \mathsf{opt}_{\mathcal{R}}(\mathcal{I})$.

However, not every solution for the convex relaxation has a corresponding solution in the original problem. Therefore, when using convex relaxations to design approximation algorithms, there is usually a *rounding* step to convert the relaxation solution to a feasible solution of the original problem. We call this procedure "rounding" because in the linear programming relaxation setting, the goal is often to "round" the fractional assignments to the variables to integral values. However, when using other relaxation techniques such as semidefinite programs, one may have to convert vector-valued variables to integral values.

Formally, a *rounding scheme* is an algorithm that takes the problem instance $\mathcal{I}$ and the optimal solution x^* to the convex relaxation $\mathcal{R}$ as input, and outputs a solution x to the original problem. Let $\mathsf{val}_{\mathcal{R}}(\mathcal{I}; x^*) = \mathsf{opt}_{\mathcal{R}}(\mathcal{I})$ be the objective value of x^*; and let $\mathsf{val}(\mathcal{I}; x)$ be the objective value of x. Now still assume that the original problem is a maximization problem (where the case of minimization problem can be similarly deduced). If the rounding algorithm can be proved to always output an x such that

$$\mathsf{val}(\mathcal{I}; x) \geq \alpha \cdot \mathsf{opt}(\mathcal{I}) \tag{1.1}$$

for some $\alpha \in [0, 1]$, then we get an α-approximation algorithm by first solving the convex relaxation and then performing the rounding scheme. However, directly proving (1.1) is often quite difficult because computing $\mathsf{opt}(\mathcal{I})$ itself is NP-hard and we do not know much about it. Alternatively we turn to prove

$$\mathsf{val}(\mathcal{I}; x) \geq \alpha \cdot \mathsf{val}_{\mathcal{R}}(\mathcal{I}; x^*) \tag{1.2}$$

which is usually easier and implies (1.1) since $\mathsf{val}_{\mathcal{R}}(\mathcal{I}; x^*) = \mathsf{opt}_{\mathcal{R}}(\mathcal{I}) \geq \mathsf{opt}(\mathcal{I})$.

1.2.3 Integrality gaps and limitations of the relaxation

As mentioned before, not every solution for the convex relaxation has a corresponding solution in the original problem. Therefore, the relaxation optimum might be way "better" than the optimum of the original problem.

We call $\mathcal{I}$ be the *integrality gap instance* with *gap ratio* $\frac{\mathsf{opt}(\mathcal{I})}{\mathsf{opt}_{\mathcal{R}}(\mathcal{I})}$. When this ratio is far from 1, $\mathcal{I}$ is a *hard instance* for the relaxation.

We also define the *integrality gap ratio* of $\mathcal{R}$ be the worst ratio between $\mathsf{opt}(\mathcal{I})$ and $\mathsf{opt}_{\mathcal{R}}(\mathcal{I})$, i.e. for maximization problems, let

$$\mathrm{IG}(\mathcal{R}) \stackrel{\mathrm{def}}{=} \inf_{\mathcal{I}} \frac{\mathsf{opt}(\mathcal{I})}{\mathsf{opt}_{\mathcal{R}}(\mathcal{I})} \leq 1.$$

Then we have

$$\mathrm{IG}(\mathcal{R}) \cdot \mathsf{opt}_{\mathcal{R}}(\mathcal{I}) \leq \mathsf{opt}(\mathcal{I}) \leq \mathsf{opt}_{\mathcal{R}}(\mathcal{I}).$$

When $\mathrm{IG}(\mathcal{R})$ is close to 1 (from below), we see that the relaxation optimum is a good estimation of the optimum of the original problem.

Similarly, for minimization problems, we let the integrality gap ratio of a specific relaxation $\mathcal{R}$ be

$$\mathrm{IG}(\mathcal{R}) \stackrel{\mathrm{def}}{=} \sup_{\mathcal{I}} \frac{\mathsf{opt}(\mathcal{I})}{\mathsf{opt}_{\mathcal{R}}(\mathcal{I})} \geq 1,$$

and we have

$$\mathsf{opt}_{\mathcal{R}}(\mathcal{I}) \leq \mathsf{opt}(\mathcal{I}) \leq \mathrm{IG}(\mathcal{R}) \cdot \mathsf{opt}_{\mathcal{R}}(\mathcal{I}).$$

The relaxation optimum well approximates the optimum of the original problem when $\mathrm{IG}(\mathcal{R})$ is close to 1 from above.

Integrality gaps serve as a measure of the quality of the relaxation $\mathcal{R}$. When the integrality gap of $\mathcal{R}$ is bad (i.e. far from 1), since we usually use (1.2) to prove the effectiveness of $\mathcal{R}$-based approximation algorithms, we tend not to get a good approximation algorithm. In this sense, designing the integrality gap instance with gap ratio far from 1 is a way to show the limitation of the relaxation.

On the other hand, whenever we establish (1.2), we also know that the integrality gap of relaxation $\mathcal{R}$ is no worse than α. This is also the usual way of proving the good quality of the relaxation.

We summarize and extend the definitions made above as follows.

Definition 1.2.1. *Fix an optimization problem Q and an LP/SDP relaxation $\mathcal{R}$ for Q. An instance $\mathcal{I}$ of Q is said to be a (c, s)-*integrality gap instance *for $\mathcal{R}$,*

- *(for maximization problems) if the optimal value of $\mathcal{R}(\mathcal{I})$ is at least c while $\mathsf{val}(\mathcal{I})$ is less than s;*
- *(for minimization problems) if the optimal value of $\mathcal{R}(\mathcal{I})$ is at most c while $\mathsf{val}(\mathcal{I})$ is more than s.*

We also say that $\mathcal{I}$ is an integrality gap instance with gap ratio $\frac{c}{s}$.

Given such an instance $\mathcal{I}$, we say that $\mathcal{R}$ has an integrality gap (c, s). *The* integrality gap ratio of $\mathcal{R}$ *(namely $\mathrm{IG}(\mathcal{R})$) is the infimum of the gap ratios of all integrality gap instances for maximization problems, and the supremum of the ratios for minimization problems.*

1.3 Linear and semidefinite programming relaxations, and methods of designing them

A large number of approximation algorithms use a specific type of convex relaxation – linear programming (LP). While linear programs can be solved in polynomial time using interior point methods [6, 225, 224], the simplex method is used extensively in practice. The "basic linear programming relaxation" (i.e. the one derived from the simple exemplary approach described in Section 1.2.1) already succeeds in efficiently approximating problems such as VERTEXCOVER, SETCOVER, and a wide class of generalized covering problems [152]. There are also ways to add more constraints (and variables if necessary) to derive stronger linear programming relaxations for problems such as MULTICUT [57, 83], SPARSESTCUT [163], and MULTIWAYCUT [56, 168, 51, 206].

Semidefinite programming (SDP) relaxations are another class of powerful convex relaxations. In a semidefinite program, the variables are vector valued, while both the constraints and the objective are linear in terms of the inner products of the variables. Semidefinite programs can be solved in polynomial time using the interior point methods [6, 225, 224]. More precisely, these algorithms output a solution with the value which differs from the optimum by at most an additive error ϵ in time that is polynomial in the program description size and $\log \frac{1}{\epsilon}$. Semidefinite programming relaxations proved to be extremely successful in approximation algorithms design after being introduced by Goemans and Williamson [94] in the context of the MAXCUT problem. [1] A few examples on the semidefinite programming relaxation-based approximation algorithms include the

[1]Indeed, the classic work by Lovász [165], known as the Lovász Theta function today, is essentially a semidefinite programming relaxation for the INDEPENDENTSET problem.

ones for constraint satisfaction problems [94, 89, 132, 227, 229, 3, 91, 59, 189], UNIFORMSPARSESTCUT [21], ordering problems [61], and discrete optimization problems [63, 9].

1.3.1 Relaxation hierarchies and the implication of their limitations

In this thesis, we are mainly interested in LP/SDP-based algorithms when talking about convex relaxations. When designing LP/SDP-based algorithms, one usually use the most natural and simple LP/SDP relaxation of the integer programming formulation of the original problem. We refer to these relaxations as *basic* LP/SDP relaxations. As we have defined before, fix a combinatorial optimization problem and a convex relaxation, the integrality gap serves as a measure of the quality of the relaxation, which is a different quantity from the hardness of approximation factor for the problem. However, for many important problems, the integrality gaps of even the simplest basic LP/SDP relaxations interestingly correspond the hardness of approximation results [189, 152, 168].

On the other hand, for some other problems, in order to strengthen the algorithmic power, one can add additional constraints into the basic relaxation, so that the resulting relaxation is tighter and gives better approximation guarantee. One notable example is the work by Arora, Rao, and Vazirani [21] which added the so-called "ℓ_2^2-triangle inequalities" to the basic SDP relaxation for UNIFORMSPARSESTCUT and improved the approximation ratio from $\Theta(\log n)$ to $O(\sqrt{\log n})$ where n is the size of the input graph.

While analysis of the convex relaxations with such extra constraints are very problem specific, there are several systematic ways to add additional constraints without even looking at the problem. Such systematic sets of constraints include the ones defined by Lovász and Schrijver [166], and Sherali and Adams [207] for LP, and the one defined by Parrilo and Lasserre [185, 157] for SDP. There are also hierarchies with a mixture of linear and semidefinite constraints, such as the Sherali-Adams+SDP relaxation hierarchy. In each of these ways, we obtain a sequence of increasingly powerful relaxations, which we often refer to as the *hierarchy* of convex relaxations. The relaxation at the r-th level (also called round) in the hierarchy typically has $n^{O(r)}$ additional constraints (and auxiliary variables), and can be solved in $n^{O(r)}$ time. Please refer to [66] for a more comprehensive introduction and comparison of these relaxation hierarchies.

Among these relaxation hierarchies, the Parrilo–Lasserre SDP hierarchy is the most powerful. Most of the known LP/SDP relaxation-based algorithms can be derived from at most the 4th level of the hierarchy (including the Arora-Rao-Vazirani algorithm for UNIFORMSPARSESTCUT). Given this, it is natural to study the limitations of the Parrilo–

Lasserre hierarchy for optimization problems where no concrete hardness of approximation result is proven. In this thesis, we will prove several new integrality gap results for the Parrilo–Lasserre hierarchy for the problems where no concrete NP-hardness result is known. Such results will help us understand the complexity of approximating those problems.

Prior to our work, several of the known results on strong integrality gap results for many rounds of the Parrilo–Lasserre hierarchy, starting with the remarkable construction by Grigoriev [97, 99] and Schoenebeck [204], apply in situations where a corresponding NP-hardness result is already known. Thus they are not "prescriptive" of hardness. In fact, besides the results introduced later in this thesis, we are aware of only the following examples where a polynomial-round Lasserre integrality gap stronger than the corresponding NP-hardness result is known: MAXk-CSP and k-COLORING [220]. Indeed, for all we know 4 rounds of the Parrilo–Lasserre hierarchy could improve the Goemans-Williamson algorithm for MAXCUT, and therefore refute the famous Unique Games Conjecture [136, 141] (please refer to Section 2.3 for more on the conjecture).

A good portion of this thesis is devoted to the study of the power of the Parrilo–Lasserre SDP hierarchy. We will be using a crucial view of the Parrilo–Lasserre SDP hierarchy as the Sum-of-Squares algebraic proof system, and therefore also call the hierarchy as SOS hierarchy. Depending on the context, we will use Parrilo–Lasserre, Lasserre, SOS, and SOS/Lasserre interchangeably throughout the thesis.

1.4 A brief overview of contributions

A large part of this thesis is devoted to the study of the power the convex (LP and SDP) relaxation hierarchies – we explore both the effectiveness and limitations of this algorithmic framework. The remaining part of the thesis consists of a collection of approximation algorithms and hardness of approximation results, with an emphasis on the design of robust algorithms.

In this section, we briefly introduce the results included in this thesis. A more detailed list on these results can be found in Chapter 3.

In Part I, we start off by showing the effectiveness of the Sherali-Adams LP relaxation hierarchy. Even given that the hierarchy does not use the power of semidefniteness, we show that Sherali-Adams LP gives the state-of-the-art approximation guarantee for a large class of problems including the dense (and locally dense) constraint satisfaction problems and assignment problems.

Then we turn to study the limitations of the Parrilo–Lasserre hierarchy. We focus on the problems such as DENSEkSUBGRAPH, BALANCEDSEPARATOR, and UNIFORMSPARSESTCUT; and prove the Parrilo–Lasserre hierarchy integrality gaps which beats the known NP-hardness of approximation results. As mentioned in Section 1.3, such type of results are rare in literature and serve as an evidence that the problems are indeed hard to approximate.

To motivate the contributions in Part II, we would like to mention another class of combinatorial optimization problems, including UNIQUEGAMES and MAXCUT. These problems seem to be significantly easier than DENSEkSUBGRAPH; however, the NP-hardness of approximation and the algorithmic bounds still do not match. While it is a general belief that DENSEkSUBGRAPH is indeed very hard to approximate, whether the NP-hardness results for UNIQUEGAMES and MAXCUT can be improved (or even an evidence for that) is of great interest in the field of approximations. Researchers have shown Sherali-Adams+SDP integrality gaps for these problems, which serve as the best evidence for the hardness of approximation. These integrality gap instances are the "hardest instances" in literature in the sense that they are resistant to the strong relaxation hierarchy (and perhaps the strongest excluding Parrilo–Lasserre). A natural question arise here is whether these gap instances are also resistant to the Parrilo–Lasserre hierarchy. An affirmative answer to this question would further consolidate our best evidence.

In Part II, however, we show that these instances turn out to be easy for the Parrilo–Lasserre hierarchy, giving a negative answer to the question. This result is obtained by viewing the Parrilo–Lasserre hierarchy from a different perspective, namely as an algebraic proof system, instead of as a semidefinite programming. While this connection was brought up by Parrilo [185] and Lasserre [156, 157] more than a decade ago, we first make use of it in the setting of combinatorial optimization problems. Our results are the first to separate the power of Parrilo–Lasserre from other hierarchies on UNIQUEGAMES and MAXCUT and seriously question the possible optimality of the state-of-the-art algorithms for the two problems. We also hope that our proof techniques help to extend our (limited) understanding of the Parrilo–Lasserre hierarchy.

In the rest of this thesis, Part III consists of robust algorithms in different settings, such as a special class of constraint satisfaction problems (the ones with width-1), MAXBISECTION, and isomorphism detection for trees. Part IV is a collection of other approximation algorithms and hardness of approximation results. Please refer to Chapter 3 for a more detailed list of these results.

Chapter 2

Preliminaries

2.1 Problems studied in this thesis

In this section, we introduce the optimization problems studied in the thesis.

2.1.1 Constraint satisfaction problems

In a constraint satisfaction problem (CSP) with arity k and alphabet set Σ, there is a set V of n variables, and a list of m constraints, where each variable takes value from an finite set of alphabet Σ, while each constraint involves exactly k variables (or at most k variables). We also refer to this problem as kCSP.

We often talk about the special cases of a CSP where each constraint in the problem is from one of the several prefixed forms. For example, when the alphabet set is $\{\mathsf{true}, \mathsf{false}\}$, we can define the following CSPs.

- In 2-SAT, each constraint is of one of the forms $v_i \vee v_j$, $\bar{v}_i \vee v_j$, $v_i, \vee \bar{v}_j$, $\bar{v}_i \vee \bar{v}_j$.
- In MAXCUT, each constraint is of one of the forms $v_i \neq v_j$.
- In 3-SAT, each constraint is of one of the forms $v_i \vee v_j \vee v_k$, $v_i \vee v_j \vee \bar{v}_k$, $v_i \vee \bar{v}_j \vee v_k$, $v_i \vee \bar{v}_j \vee \bar{v}_k$, $\bar{v}_i \vee v_j \vee v_k$, $\bar{v}_i \vee v_j \vee \bar{v}_k$, $\bar{v}_i \vee \bar{v}_j \vee v_k$, $\bar{v}_i \vee \bar{v}_j \vee \bar{v}_k$.
- In 2-LIN, each constraint is of one of the forms $x_i \oplus x_j$, $\overline{x_i \oplus x_j}$.
- In 3-LIN, each constraint is of one of the forms $x_i \oplus x_j \oplus x_k$, $\overline{x_i \oplus x_j \oplus x_k}$.

- In HORN3-SAT, each constraint is of one the the forms $x_i, \bar{x}_i, x_i \to x_j, x_i \wedge x_j \to x_k$.

When the alphabet set becomes $\{0, 1, 2, \ldots, q-1\}$, the following CSPs are also often used in this thesis.

- In 2-LIN($\mathbb{Z}_q$), each constraint is of one of the forms $x_i \pm x_j = c \mod q$ (for some $c \in \{0, 1, 2, \ldots, q-1\}$).
- In 3-LIN($\mathbb{Z}_q$), each constraint is of one of the forms $x_i \pm x_j \pm x_k = c \mod q$ (for some $c \in \{0, 1, 2, \ldots, q-1\}$).
- In Γ-2-LIN($\mathbb{Z}_q$), each constraint is of one of the forms $x_i - x_j = c \mod q$ (for some $c \in \{0, 1, 2, \ldots, q-1\}$).
- In Γ-3-LIN($\mathbb{Z}_q$), each constraint is of one of the forms $x_i + x_j - x_k = c \mod q$ (for some $c \in \{0, 1, 2, \ldots, q-1\}$).

Given a CSP Γ, the natural optimization task, called "MAXΓ", is to find an assignment to the variables such that the total weight of the satisfied constraints is maximized, where we assume that each constraint has a nonnegative weight, and the sum of all weights are 1. Fix a CSP instance $\mathcal{I}$, we use $\mathsf{val}(\mathcal{I})$ to denote the value of the optimal solution. We say that $\mathcal{I}$ is "satisfiable" if $\mathsf{val}(\mathcal{I}) = 1$.

2.1.1.1 The UNIQUEGAMES problem

The UNIQUEGAMES (UG for short) problem is a special binary CSP with domain size q. For each constraint e, there is a bijection π_e attached to it. The constraint e is satisfied when the two corresponding variables x_i and x_j take values so that $\pi_e(x_i) = x_j$. The goal is to find an assignment to the variables so that the fraction of satisfied constraints is maximized. The famous Unique Games Conjecture informally states that UG is very hard to approximate, we will discuss a little more about the conjecture in Section 2.3.

2.1.2 Graph partitioning problems

Partitioning a graph into two (balanced) parts with few edges going across them is a fundamental optimization problem. Graph partitions or separators are widely used in many applications (such as clustering, divide and conquer algorithms, VLSI layout, etc). In this thesis, we will focus on the following prototypical objectives of graph partitioning.

Definition 2.1.1 (BALANCEDSEPARATOR). *Given an undirected graph $G = (V, E)$ and $0 < \tau < .5$, the goal of the τ vs. $1 - \tau$* BALANCEDSEPARATOR *problem is to find a set $A \subseteq V$ such that $\tau|V| \leq |A| \leq (1 - \tau)|V|$, while* $\mathrm{edges}(A, V \setminus A)$ *is minimized. Here* $\mathrm{edges}(A, B)$ *is the number of edges in E that cross the cut (A, B).*

Definition 2.1.2 (UNIFORMSPARSESTCUT). *Given an undirected graph $G = (V, E)$, the goal of the* UNIFORMSPARSESTCUT *problem is to find a set $\emptyset \subsetneq A \subsetneq V$ such that the sparsity*

$$\frac{\mathrm{edges}(A, V \setminus A)}{|A||V \setminus A|}$$

is minimized.

In some other cases, we are also interested in finding a partition that maximizes the number of edges those cross the partition (such as MAXCUT). In addition, we define the following problem which is MAXCUT plus a global cardinality constraint.

Definition 2.1.3 (MAXBISECTION). *Given an undirected graph $G = (V, E)$, the goal of the* MAXBISECTION *problem is to find a set $A \subseteq |V|$ such that $|A| = \frac{1}{2}|V|$ (assume that $|V|$ is an even number), such that*

$$\frac{\mathrm{edges}(A, V \setminus A)}{|E|}$$

is maximized.

We may also use the density of the edges within a subset as the objective value (instead of counting the number of edges across the partition). In the light of this, we define

Definition 2.1.4 (DENSEkSUBGRAPH). *Given an undirected graph $G = (V, E)$ and an integer k, the goal of the* DENSEkSUBGRAPH *problem is to find a subset $A \subseteq V$ such that $|A| = k$ and number of edges within A is maximized.*

2.1.3 Graph isomorphism and assignment problems

The GRAPHISOMORPHISM problem is one of the most intriguing and notorious problems in computational complexity (we will also refer to it as GISO for short); we refer to [149, 30, 24, 148, 69] for surveys. Together with FACTORING, it is one of the very rare problems in NP which is not known to be in P but which is believed to be not NP-hard [29, 49, 205] (according to standard complexity-theoretic assumptions).

The most well-known heuristic for GISO is the Weisfeiler-Lehman (WL) algorithm [222] and its "higher dimensional" generalizations. These heuristics, given two graphs, are always correct when the two graphs are isomorphic, but might misreport the nonisomorphic pairs of graphs as being isomorphic. The "k-dimensional generalization" WL^k (see [222, 55] for discussion) runs in time $n^{O(k)}$ and is more and more powerful as k grows larger (i.e. misreports less nonisomorphic pairs of graphs as being isomorphic). The WL^k heuristic is very powerful. For example, it is known to work correctly in polynomial time for all graphs which exclude a fixed minor [102], a class which includes all graphs of bounded tree width or bounded genus. Spielman's $2^{\tilde{O}(n^{1/3})}$-time graph isomorphism algorithm [212] for strongly regular graphs is achieved by WL^k with $k = \tilde{O}(n^{1/3})$. The WL^k algorithm with $k = O(\sqrt{n})$ is also a key component in the $2^{O(\sqrt{n \log n})}$-time GISO algorithm [33]. Throughout the '80s there was some speculation that GISO might be solvable on all graphs by running the WL^k algorithm with $k = O(\log n)$ of even $k = O(1)$. However this was disproved in the notable work of Cai, Früer, and Immerman [55], which showed the existence of pairs of nonisomorphic n-vertex graphs which are not distinguished by WL^k unless $k = \Omega(n)$.

In this thesis proposal, we will study a potentially stronger algorithmic framework than the WL algorithm, and also study the GISO problem from the approximation algorithms prospect and using tools from approximation algorithms design. In order to do this, we introduce several new definitions here.

Definition 2.1.5. *Let G and H be nonempty n-vertex graphs. For $0 \leq \beta \leq 1$, we say that a permutation $\pi : V(G) \to V(H)$ is an α-isomorphism if*

$$\frac{|\{(u,v) \in E(G) : (\pi(u), \pi(v)) \in E(H)\}|}{\max\{|E(G)|, |E(H)|\}} \geq \alpha,$$

where $V(G)$ and $V(H)$ are the vertex sets of G and H respectively, and $E(G)$ and $E(H)$ are the edge sets of G and H respectively.

If there exists an α-isomorphism between G and H, we say that G and H are α-isomorphic.

Observe that this definition is symmetric in G and H. The two graphs are isomorphic if and only if they are 1-isomorphic. The classical GISO problem is to check whether the two input graphs are 1-isomorphic. Now we introduce the following natural optimization version of the problem.

Definition 2.1.6 (MAXGISO)**.** *Given two n-vertex graphs G and H, the* MAXGISO *problem is to find a permutation $\pi : V(G) \to V(H)$ such that π is an α-isomorphism and α is maximized.*

The QUADRATICASSIGNMENTPROBLEM (QAP for short) is a natural generalization of the MAXGISO problem. We now define the even more generalized problem kAP, where the QAP problem is a special case when $k = 2$, as follows.

Definition 2.1.7 (kAP)**.** *For an integer $k \geq 2$, an instance of the degree k assignment problem (kAP) is given as $\mathcal{I} = (V, \omega)$, where V is the set of variables, ω is a distribution over $V^k \times V^k$. The goal is find a permutation π of V such that the value of π, defined as*

$$\mathsf{val}(\mathcal{I}, \pi) = n^k \Pr_{(U,W)\sim\omega} \left[\forall i \in \{1, 2, 3, \ldots, k\} : \pi(u_i) = w_i\right],$$

is maximized. $U = (u_1, u_2, \ldots, u_k)$ and $W = (w_1, w_2, \ldots, w_k)$. We define the optimal value of $\mathcal{I}$ to be $\mathsf{val}(\mathcal{I}) = \max_\pi \{\mathsf{val}(\mathcal{I}, \pi)\}$.

2.1.4 Dense and locally-dense instances

Given a CSP instance $\mathcal{I}$ with arity k and variable set V, let $\omega : V^k \to \mathbb{R}$ be the weights on the constraints. I.e. let $\omega(v_1, v_2, \ldots, v_k)$ be the weight on the constraint imposed on the k-tuple $(v_1, v_2, \ldots, v_k)$; let $\omega(v_1, v_2, \ldots, v_k)$ be 0 if there is no such constraint. Since we assumed that the weights are nonnegative and sum up to 1, we can view ω as a probability distribution on V^k. We say the CSP instance $\mathcal{I}$ is Δ-dense if ω is Δ-dense; say $\mathcal{I}$ is Δ-locally dense if ω is Δ-locally dense. We also say an instance is dense or locally-dense if it is $O(1)$-dense or $O(1)$-locally dense. Here we define,

Definition 2.1.8 (dense and locally dense distributions)**.** *Let ω be a probability distribution over a finite set Ω. For $\Delta \in (0, 1]$, we say ω is Δ-dense if for every $a \in \Omega$, it holds that $\Delta \cdot \omega(a) \leq \frac{1}{|\Omega|}$.*

Let ω be a probability distribution over V^k. Let $d_i(v) = \Pr_{S\sim\omega}[S_i = v_i]$ be the probability that the i-th coordinate is v under ω. For $\Delta \in (0, 1]$, we say ω is Δ-locally dense *if for every $(v_1, \ldots, v_k) \in V^k$, it holds that*

$$\Delta \cdot \omega(v_1, \ldots, v_k) \leq \frac{1}{|V|^{k-1}} \sum_{1\leq i\leq k} d_i(v_i).$$

Since $d_i(v) = \sum_{S\in V^k : S_i = v} \omega(S)$, the RHS of the locally dense condition is equal to $\sum_{1\leq i\leq k} \mathop{\mathbf{E}}_{S\sim V^k} [\omega(S) \mid S_i = v_i]$. Thus the locally dense condition says that no tuple $(v_1, \ldots, v_k)$ is "wild" in that $\omega(v_1, \ldots, v_k)$ is at most constant times the sum over i of the average probability mass of S with $S_i = v_i$.

The notion of local density is introduced in [75] to generalize the metric condition. To see this, suppose $\omega : V^2 \to \mathbb{R}$ is a metric. Then, ω is 1-locally dense since, for any $u, v \in V$, we have $\frac{1}{n}(d_1(u) + d_2(v)) = \frac{1}{n}\sum_w(\omega(u, w) + \omega(w, v)) \geq \frac{1}{n}\sum_w \omega(u, v) \geq \omega(u, v)$.

2.2 Approximation and hardness of approximation

In this section, we define the notions of approximation algorithms and hardness of approximation.

Let us first fix a problem Q (which is usually computationally hard to calculate the exact optimal objective solution). We suppose that the problem Q is a maximization problem throughout this subsection, while the definitions given can be naturally adapted to minimization problems.

Fix an instance $\mathcal{I}$ from the problem Q. We denote the optimal objective value of the problem to be $\mathsf{val}(\mathcal{I})$. Given an algorithm A for the problem G, we use $\mathsf{val}_A(\mathcal{I})$ to denote the value of the solution output by A on input G. Now we define the following measure of the quality of A based on the approximation ratio.

Definition 2.2.1. *We say that an algorithm A is an α-approximation algorithm ($0 \leq \alpha \leq 1$) for the problem Q if for every instance $\mathcal{I}$ from Q, we have*

$$\frac{\mathsf{val}_A(\mathcal{I})}{\mathsf{val}(\mathcal{I})} \geq \alpha.$$

We also introduce the definition of polynomial-time approximation scheme (PTAS) where the algorithm gives arbitrarily (and constantly) good approximation to the problem in polynomial time.

Definition 2.2.2. *Fix an optimization problem Q, a PTAS is an algorithm which takes an instance $\mathcal{I}$ of Q of size n and a parameter $\epsilon > 0$, and in time $T(\epsilon, n)$, outputs a solution that is $(1-\epsilon)$-approximation to the optimal solution of the instance, where for every $\epsilon > 0$ there exists a constant $C = C(\epsilon)$ such that $T(\epsilon, n) \leq O(n^C)$.*

Similarly, a quasi-polynomial-time approximation scheme (quasi-PTAS) for Q is an algorithm which takes an instance $\mathcal{I}$ of Q of size n and a parameter $\epsilon > 0$, and in time $T'(\epsilon, n)$, outputs a solution that is $(1-\epsilon)$-approximation to the optimal solution of the instance, where for every $\epsilon > 0$ there exists a constant $C' = C'(\epsilon)$ such that $T(\epsilon, n) \leq 2^{C(\log n)^{O(1)}}$.

Sometimes we need a more refined definition than Definition 2.2.1. Take the problem MAXCUT for example (please refer to Section 2.1.1 for the precise definition of MAXCUT). Currently the best known approximation algorithm for MAXCUT is by Goemans and Williamson [94]; and by Definition 2.2.1, the Goemans-Williamson algorithm is an α_{GW}-approximation algorithm (where $\alpha_{GW} \approx .878$). However, when the optimal solution in a MAXCUT instance cuts almost all the edges (say $(1-\epsilon)$ of the edges), Goemans-Williamson algorithm guarantees to output a cut that cuts $(1-O(\sqrt{\epsilon}))$ of the edges, while we expect a $.878$-approximation algorithm to output a cut only cutting $.878(1-\epsilon)$ of the edges (which is much smaller than $(1-O(\sqrt{\epsilon}))$ when ϵ is small). Therefore, we need to introduce the following refined notion of approximation to address this difference.

Definition 2.2.3. *Fix $c \geq s \geq 0$, we say that an algorithm A is a (c,s)-approximation algorithm for the problem Q if for every instance $\mathcal{I}$ from Q, when $\mathsf{val}(\mathcal{I}) \geq c$, we have $\mathsf{val}_A(\mathcal{I}) \geq s$.*

By definition Definition 2.2.3, an α-approximation algorithm is a $(c, \alpha c)$-approximation algorithm for every $c > 0$. The Goemans-Williamson algorithm is an α_{GW}-approximation algorithm for MAXCUT in general; but it is also a $(1-\epsilon, 1-O(\sqrt{\epsilon}))$-approximation algorithm.

Definition 2.2.3 motivates us to define the following decision problem for every optimization problem Q.

Definition 2.2.4. *Given an optimization problem Q, for every $c \geq s > 0$, let the problem (c,s)-gap-Q be the problem that given an instance $\mathcal{I}$, to*

- *output YES when $\mathsf{val}(\mathcal{I}) \geq c$;*
- *output NO when $\mathsf{val}(\mathcal{I}) < s$.*

A simple observation is that fix the problem Q and the parameters $c \geq s > 0$, if (c,s)-gap-Q is NP-hard, then it is NP-hard to $\frac{c}{s}$-approximate the problem Q. Therefore, a usual strategy of proving hardness of approximation statement for an optimization problem Q is to prove the hardness of (c,s)-gap-Q problem.

2.2.1 Robust algorithms

Robust algorithms are approximation algorithms concerned with the case that the problem has an "almost perfect" solution (e.g. when all the constraints are satisfied, when two graphs are isomorphic, etc.). At a high level, instead of giving approximation ratio guarantee for general inputs, robust algorithms extremely well-approximate an almost perfect solution when such solutions exist. Such algorithms can be viewed as a robust version of the algorithms designed to find (exact) perfect solutions. One motivation to design robust algorithms is that in practical situations (e.g. learning with noise), instances with perfect solutions might be corrupted by a small amount of noise; an robust algorithm becomes useful since it can still satisfy most of the constraints of the noisy instance.

The notion of robust algorithms was first explicitly introduced by Zwick [228] for constraint satisfiability problems (CSPs), where he showed robust algorithms for several CSPs including MAX2SAT and MAXHORNSAT. In this proposal, we will focus on the robust algorithms for general CSPs, MAXBISECTION, and GRAPHISOMORPHISM. Now we give the explicit definitions for these robust algorithms.

Definition 2.2.5 (ROBUSTSATISFIABILITY)**.** *Fix a CSP, we say that an algorithm A is a robust satisfiability algorithm for the CSP if there exists a function $r : [0,1] \to [0,1]$ satisfying $r(\epsilon) \to 0$ as $\epsilon \to 0^+$ such that whenever A is given an instance $\mathcal{I}$ with $\mathsf{val}(\mathcal{I}) \geq 1-\epsilon$, A outputs a solution satisfying $(1 - r(\epsilon))$ of the constraints.*

By definition, the famous Goemans-Williamson [94] algorithm for MAXCUT is an robust satisfiability algorithm with $r(\epsilon) = O(\sqrt{\epsilon})$. Zwick [228] gave an robust satisfiability algorithm for MAX2SAT with $r(\epsilon) = O(\sqrt{\epsilon})$, and an robust satisfiability algorithm for MAXHORNSAT with $r(\epsilon) = O\left(\frac{\log\log(1/\epsilon)}{\log(1/\epsilon)}\right)$.

We also define the notion of robust algorithms for the following two problems which will be studied in this thesis.

Definition 2.2.6 (ROBUSTMAXBISECTION)**.** *We say that an algorithm A solves the* ROBUSTMAXBISECTION *problem if there exists a function $r : [0,1] \to [0,1]$ satisfying $r(\epsilon) \to 0$ as $\epsilon \to 0^+$ such that whenever A is given an undirected graph G with* MAXBISECTION *optimum at least $1-\epsilon$, A outputs a bisection with $(1 - r(\epsilon))$ of the edges across the bisection.*

Definition 2.2.7 (ROBUSTGISO)**.** *We say that an algorithm A solves the* ROBUSTGISO *problem if there exists a function $r : [0,1] \to [0,1]$ satisfying $r(\epsilon) \to 0$ as $\epsilon \to 0^+$ such that whenever A is given a pair of graphs which are $(1-\epsilon)$-isomorphic, A outputs a $(1 - r(\epsilon))$-isomorphism between them.*

2.3 The Unique Games Conjecture

The Unique Games Conjecture states,

Conjecture 1 (Unique Games Conjecture [136]). *For every constant $\epsilon > 0$, there exists a large enough q so that the $(1-\epsilon, \epsilon)$-gap-*UG *problem with domain size q is* NP*-hard.*

The Unique Games Conjecture, soon after its initial proposal Khot [136], became a central problem in approximation algorithms research. If the conjecture is true, it implies optimal inapproximability results for many problems including the broad class of CSPs [189], covering and packing problems [152], ordering CSPs [106] and MULTIWAYCUT [168]. While numerous research has been conducted to investigate the correctness of the conjecture (e.g. [136, 218, 144, 105, 59, 19, 16, 193]), the status of the conjecture remains a major open question in the field, and there is no compelling opinion about its truth. However, through these intensive studies of UG, many connections have emerged between the conjecture, analysis, geometry and mathematical programming, leading to many exciting advancements in both algorithm design and hardness of approximations. The readers are encouraged to refer to [138] for more on the conjecture.

Chapter 3

Summary of contributions and organization of this thesis

The technical part of this thesis consists of 4 parts.

Part I and Part II will discuss how to strengthen our current (possibly limited) understanding of convex relaxation hierarchies. Table 3.1 is a list of the relevant results.

Part III will be on the design (and impossibility of designing) robust algorithms. Part IV will consist of a few other inapproximability results. Table 3.2 is a list of the relevant results.

Finally, the thesis will be concluded by Part V with a few future directions.

In the following 4 sections, we will give overviews of the 4 technical parts respectively.

3.1 Overview of Part I: study of the LP/SDP relaxation hierarchies

As we mentioned before, the LP/SDP relaxation hierarchies are parameterized by an integer r – the level in the hierarchy. One challenge here is to understand the trade-off between the approximation guarantee and the number of levels. In this subsection, we describe the results related to this question. For some problems, we prove that a small number of levels in the hierarchies effectively approximates the optimal solution; in other cases, we show lower bounds for the hierarchies, i.e. a large number of levels is needed to obtain good approximation.

Effectiveness of hierarchies	Limitation of hierarchies
The Sherali-Adams LP relaxation hierarchy for dense and locally-dense instances (Theorem 3.1.1, Theorem 3.1.2, and Theorem 3.1.3) The Parrilo–Lasserre SDP relaxation hierarchy for known UNIQUEGAMES, BALANCEDSEPARATOR, and MAXCUT integrality gap instances (Theorem 3.2.1, Theorem 3.2.2, and Theorem 3.2.3)	Polynomial-round Parrilo-Lasserre integrality gap instances for DENSEkSUBGRAPH (Theorem 3.1.4) Linear-round Parrilo-Lasserre integrality gap instances for BALANCEDSEPARATOR and UNIFORMSPARSESTCUT (Theorem 3.1.5 and Theorem 3.1.6) Linear Parrilo-Lasserre SDP does not tell isomorphic graphs from far-from-being-isomorphic graphs (Theorem 3.1.7)

Table 3.1: Table of contributions on convex relaxation hierarchies

3.1.1 Algorithmic results

In Chapter 4, we study the algorithmic guarantee of the Sherali-Adams LP relaxation hierarchy for dense and locally dense CSPs and assignment problems (APs). Prior to our work, there was a long series of works (e.g. [74, 87, 18, 77, 17, 76, 75]), using various techniques, such as sampling, regularity lemma, and tensor decomposition, to design PTAS and quasi-PTAS for these instances.

We show that the Sherali-Adams LP relaxation hierarchy is a unified algorithmic framework to obtain all the previously known results. In particular, we first show that

Theorem 3.1.1 (Pre-statement of Theorem 4.1.1)**.** *For any $\epsilon > 0$, $O(\frac{1}{\epsilon^2})$-round Sherali-Adams LP relaxation hierarchy gives $(1-\epsilon)$-approximation to dense or locally dense* MAXkCSP.

Then, we turn to dense and locally dense MAXkCSP with global cardinality constraints. For explanatory purposes, we only consider *bisection* constraint, i.e., the domain is $\{0, 1\}$ and the number of variables that are assigned to 0 should be equal to the number of variables that are assigned to 1. We show that

Theorem 3.1.2 (Pre-statement of Theorem 4.8.2)**.** *For any $\epsilon > 0$, $O(\frac{1}{\epsilon^2})$-round Sherali-Adams LP relaxation hierarchy gives $(1-\epsilon)$-approximation to dense or locally dense*

Approximation algorithms	Hardness of approximation
Robust algorithm for width-1 constraint satisfiability problems (Theorem 3.3.1)	Hardness of ROBUSTGISO (Theorem 3.3.3)
An algorithm for ROBUSTMAXBISECTION (Theorem 3.3.2)	Hardness of MAXΓ-2-LIN and MAXΓ-3-LIN over integers (Theorem 3.4.3)
A robust isomorphism algorithm for trees (Theorem 3.3.4)	Hardness of approximating almost satisfiable MAXHORN3-SAT (Theorem 3.4.4)
Approximating the $2 \to 4$ norm of random linear operators (Theorem 3.4.1)	

Table 3.2: Table of contributions on other approximation algorithms and hardness of approximation

bisection MAXkCSP.

Finally, we consider the dense MAXkAP problems, and show that

Theorem 3.1.3 (Pre-statement of Theorem 4.1.3)**.** *For any $\epsilon > 0$, $O(\frac{\log n}{\epsilon^2})$-round Sherali-Adams LP relaxation hierarchy gives $(1 - \epsilon)$-approximation to dense or locally dense* MAXkAP *problems with n variables.*

3.1.2 Integrality gaps

In Chapter 5, Chapter 6, and Chapter 7, we study several important combinatorial optimization problems and show that the strongest known SDP hierarchy (i.e. the Parrilo–Lasserre hierarchy) does not give good approximation for them. Given that there is no concrete inapproximability result for these problems, and (as pointed out previously) that our results are among the few ones proving Parrilo–Lasserre lower bounds beating known NP-hardness results, our lower bound theorems can be viewed as strong evidence of the inapproximability of these fundamental combinatorial optimization problems.

The DENSEkSUBGRAPH **problem.** The DENSEkSUBGRAPH problem is believed to be very hard to approximate as the best known approximation algorithm due to [41] gives $O(n^{1/4+\epsilon})$-approximation in time $n^{O(1/\epsilon)}$ for any constant $\epsilon > 0$. On the inapproximability side, [88] initially showed that a small constant factor inapproximability for DENSEkSUBGRAPH using the random 3-SAT assumption. [137] used quasi-random PCPs to rule out a PTAS. More recently, [193, 7] used more non-standard assumptions to rule out any constant factor approximation algorithms. In the work with Bhaskara et al. [40], we showed the following integrality gap theorem.

Theorem 3.1.4 (Pre-statement of Theorem 5.1.1)**.** *For every $\epsilon > 0$, there is a lower bound of $n^{2/53-\epsilon}$ on the integrality gap of level-$n^{\Omega(\epsilon)}$ Parrilo–Lasserre SDP relaxation hierarchy for the* DENSEkSUBGRAPH *problem; there is also a lower bound of n^{ϵ} on the integrality gap of level-$n^{1-O(\epsilon)}$ Parrilo–Lasserre SDP relaxation hierarchy.*

The BALANCEDSEPARATOR **and** UNIFORMSPARSESTCUT **problems.** For these two problems, the best algorithms, based on semidefinite relaxations (SDPs) with triangle inequalities, give $O(\sqrt{\log n})$-approximation [21]. On the inapproximability side, a Polynomial Time Approximation Scheme (PTAS) is ruled out for both problems assuming 3-SAT does not have randomized subexponential-time algorithms [10]. In the work with Guruswami and Sinop [112], we showed the following integrality gaps for the two problems.

Theorem 3.1.5 (Pre-statement of Theorem 6.1.2)**.** *For $0.45 < \tau < 0.5$, there are linear-round Parrilo–Lasserre SDP gap instances for the τ vs $(1-\tau)$* BALANCEDSEPARATOR *problem, such that the integral optimal solution is at least $(1+\epsilon(\tau))$ times the SDP solution, where $\epsilon(\tau) > 0$ is a constant dependent on τ.*

Theorem 3.1.6 (Pre-statement of Theorem 6.1.3)**.** *There are linear-round Parrilo–Lasserre SDP gap instances for the* UNIFORMSPARSESTCUT *problem, such that the integral optimal solution is at least $(1+\epsilon)$ times the SDP solution, for some constant $\epsilon > 0$.*

The GRAPHISOMORPHISM **problem.** A recent work of Atserias and Maneva [27] (see also [103]) shows that the power of WL^k algorithm is precisely sandwiched between the k-th and $(k+1)$-st level of the canonical Sherali-Adams LP relaxation hierarchy of the GISO problem. Given the power of WL^k algorithm, this connection shows that LP relaxation hierarchies are also useful for solving GISO. On the other hand, by the work of [55], we also know that the Sherali-Adams LP relaxation hierarchy also needs linearly many levels to fully solve GISO.

This raises the natural question whether stronger LP/SDP relaxation hierarchies might prove more powerful than WL^k in the context of GISO. In the work with O'Donnell et al. [182], we study the Parrilo–Lasserre SDP relaxation hierarchy (which is the strongest known hierarchy known in the literature as discussed before) for GISO and show that

Theorem 3.1.7 (Pre-statement of Theorem 7.1.2)**.** *For infinitely many n, there exists pairs of n-vertex, $O(n)$-edge graphs G and H such that*

- *G and H are not $(1 - 10^{-14})$-isomorphic;*
- *in order to tell that G and H are not 1-isomorphic (i.e. isomorphic), the Parrilo–Lasserre SDP relaxation hierarchy needs $\Omega(n)$ levels.*

This theorem says that the linear-level Parrilo–Lasserre hierarchy not only fails on distinguishing nonisomorphic pairs of graphs, but also fails spectacularly – the hierarchy cannot tell the two graphs are nonisomorphic even when they are different by a constant fraction of the edges.

3.2 Overview of Part II: using the Parrilo–Lasserre hierarchy to solve hard instances for weaker hierarchies

In Part II, we study the Parrilo–Lasserre SDP relaxation hierarchy when applied to the known integrality gap instances (for other relaxation hierarchies such as Sherali-Adams+SDP) in literature for several central combinatorial optimization problems, and will show that these instances are no longer integrality gap instances for constant-level Parrilo–Lasserre SDP relaxation hierarchy. In order to obtain such types of results, we will use a special and novel view of the Parrilo–Lasserre SDP relaxation hierarchy, i.e. to view the hierarchy as the so-called "sum-of-squares proof system" and to prove the success of the hierarchy (on given instances) via giving a proof that the given instance does not have great objective value in the sum-of-squares proof system. Using this connection, we hope to understand more about the the power of Parrilo–Lasserre SDP relaxation hierarchy, and proof techniques that might be helpful to construct integrality gaps for the Parrilo–Lasserre hierarchy. In particular, we will present the following results along this line.

The UNIQUEGAMES problem. We begin with showing that a very small constant level of the Parrilo–Lasserre SDP hierarchy suffices to solve the UNIQUEGAMES instances in the literature.

Theorem 3.2.1 (Pre-statement of Theorem 9.0.8)**.** *For sufficiently small ϵ and large k, and every $n \in \mathbb{N}$, let $\mathcal{W}$ be an n-variable k-alphabet* UNIQUEGAMES *instance of the type considered in [192, 154, 142] obtained by composing the "quotient noisy cube" instance of [144] with the long-code alphabet reduction of [141] so that the best assignment to $\mathcal{W}$'s variables satisfies at most an ϵ fraction of the constraints. Then, there is a degree-8 SOS refutation for the statement that the best assignment to $\mathcal{W}$'s variables satisfy at least $1/100$ fraction of the constraints.*

Thus just the level-4 Lasserre SDP hierarchy (essentially) solves the the UNIQUEGAMES instances.

The BALANCEDSEPARATOR **problem.** Devanur et al [79] gave a family of n-vertex BALANCEDSEPARATOR instances (which we will refer to as the DKSV instances) which are integrality gap instances with ratio $\Theta(\log\log n)$ for the natural SDP relaxation with triangle inequalities. Raghavendra and Steurer [188] showed that a factor-$(\log\log n)^{\Omega(1)}$ gap persists for these instances even for $(\log\log n)^{\Omega(1)}$ rounds of the "LH+SDP relaxation hierarchy". In the work with O'Donnell [184], we show that

Theorem 3.2.2 (Corollary of Theorem 10.3.1 and Theorem 10.3.3)**.** *The level-2 Parrilo–Lasserre SDP relaxation hierarchy for the* BALANCEDSEPARATOR *problem has integrality gap at most $O(1)$ for the DKSV instances.*

The MAXCUT **problem.** Assuming the Unique Games Conjecture, Khot et al. [141] showed that the Goemans-Williamson algorithm [94] for MAXCUT achieves the best possible approximation factor, namely $\alpha_{GW} \approx .878$ approximation. Khot and Vishnoi [144] gave integrality gap instances of ratio α_{GW} for the MAXCUT problem, by composing their UNIQUEGAMES instances with the MAXCUT reduction in [141]. Khot and Saket [154] subsequently showed that this gap persists even for level-$(\log\log\log n)^{\Omega(1)}$ Sherali-Adams+SDP relaxation hierarchy. In the work with O'Donnell [184], we show that constant level of the Parrilo–Lasserre SDP relaxation hierarchy gives better than α_{GW}-approximation to the integrality gap instances by Khot and Vishnoi. In particular, we prove that

Theorem 3.2.3 (Pre-statement of Theorem 8.1.3)**.** *There exists a universal integer constant C such for the level-C Parrilo–Lasserre SDP relaxation hierarchy, the Khot-Vishnoi* MAXCUT *instance has integrality gap ratio at most $1/.952 (< 1/\alpha_{GW})$.*

3.3 Overview of Part III: robust algorithms

In this subsection, we present several of our related results on robust algorithms.

Robust algorithms for satisfiability problems. As mentioned previously, Zwick [228] showed robust algorithms for MAX2SAT and MAXHORNSAT. In sharp contrast, for other problems such as MAX3LIN, although deciding whether an instance is satisfiable is in P, there is no efficient robust algorithm unless $\mathsf{P} = \mathsf{NP}$ [116]. A natural theoretical question arising at this point is to characterize the class of CSPs that admit efficient robust algorithms.

Towards answering this question, in a work with Guruswami [114], we conjectured that the CSPs which have efficient robust algorithms (assuming $\mathsf{P} \neq \mathsf{NP}$) are precisely those of "bounded width" – a notion frequently used in algebraic dichotomy theory where people study the characterization of CSPs with exact satisfiability algorithms (we will refer to this conjecture as Guruswami–Zhou Conjecture throughout this thesis). Roughly speaking, bounded-width CSPs are the ones do not encode linear equations over abelian groups; they also coincide with the CSPs solvable by the "k-consistency heuristic" algorithm in artificial intelligence. If our conjecture is true, the natural basic SDP relaxation would be the desired robust algorithm for every bounded-width CSP.

Towards proving the conjecture, in Chapter 12, we prove the following theorem.

Theorem 3.3.1 (Pre-statement of Theorem 12.1.1)**.** *If a CSP has "width-*1*", there is a polynomial-time robust satisfiability algorithm for the CSP; and the algorithm is based on the natural LP relaxation of the problem.*

The Guruswami–Zhou Conjecture was later fully confirmed by Barto and Kozik [38].

The ROBUSTMAXBISECTION **problem.** In Chapter 13, we will prove the first polynomial-time algorithm for the ROBUSTMAXBISECTION problem. Our theorem is stated as follows.

Theorem 3.3.2 (Pre-statement of Theorem 13.1.3)**.** *There is a randomized polynomial-time algorithm such that for every* $\epsilon > 0$*, given an edge-weighted graph* G *with a* MAXBISECTION *solution of value*[1] $(1-\epsilon)$*, finds a* MAXBISECTION *of value* $(1 - O(\sqrt[3]{\epsilon}\log(1/\epsilon)))$.

[1]The value of a cut in an edge-weighted graph is defined as the weight of the edges crossing the cut divided by the total weight of all edges.

Prior to our work ([107]), researchers from theoretical computer science and operations research designed various algorithms for this problem [93, 226, 115, 91], but none of them is guaranteed to find a bisection cutting most of the edges even when the graph has a near-perfect bisection cutting $(1-\epsilon)$ of the edges (in fact, they may not even cut 75% of the edges).

The ROBUSTGISO **problem.** Although it is not known whether GISO has polynomial-time algorithm, we show that assuming the so-called Feige's R3XOR hypothesis [88], there is no polynomial-time algorithm for ROBUSTGISO. In particular, we prove the following theorem.

Theorem 3.3.3 (Pre-statement of Theorem 7.1.4)**.** *Assume Feige's* R3XOR *Hypothesis [88]. Then there is no polynomial-time algorithm for* ROBUSTGISO*. More precisely, there exists $\epsilon_0 > 0$, such that suppose there exists $\epsilon > 0$ and a $t(n)$-time algorithm which can distinguish $(1-\epsilon)$-isomorphic n-vertex, m-edge graph pairs from pairs which are not even $(1-\epsilon_0)$-isomorphic (where $m = O(n)$). Then there is a universal constant $\Delta \in \mathbb{Z}^+$ and a $t(O(n))$-time algorithm which outputs "typical" for almost all n-variable, Δn-constraint instances of the* 3-XOR *problem, yet which never outputs "typical" on instances which are $(1-\Theta(\epsilon))$-satisfiable.*

Our proof uses a linear-size reduction from ROBUSTGISO to MAX3-LIN and uses a hardness of approximation result for MAX3-LIN by Håstad [116]. Therefore, by the efficient-construction of the PCP theorem [173], and assuming the ETH ([124], i.e. 3-SAT does not have subexponential-time algorithms), there is no subexponential-time algorithm for ROBUSTGISO. This is in contrast with the $2^{O(\sqrt{n \log n})}$-time algorithm for GISO due to [33].

Despite the impossibility of constructing efficient ROBUSTGISO algorithm for general graphs, we design a ROBUSTGISO algorithm for trees. In Chapter 14, we prove

Theorem 3.3.4 (Pre-statement of Theorem 14.1.5)**.** *Given two n-vertex trees T_1 and T_2, there is a polynomial-time algorithm to find a $\left(1 - O\left(\epsilon^{1/4}\right)\right)$-isomorphism between them whenever they are $(1-\epsilon)$-isomorphic, for every $\epsilon > 0$.*

3.4 Overview of Part IV: other approximation and hardness of approximation results

In this subsection, we describe several related hardness of approximation results obtained by the author.

3.4.1 Approximating the $2 \to 4$ norm of random linear operators

For a function $f\colon \Omega \to \mathbb{R}$ on a (finite) probability space Ω, the *p-norm* is defined as $\|f\|_p = (\mathbf{E}_\Omega f^p)^{1/p}$. The $p \to q$ *norm* $\|A\|_{p\to q}$ of a linear operator A between vector spaces of such functions is the smallest number $c \geq 0$ such that $\|Af\|_q \leq c\,\|f\|_p$ for all functions f in the domain of A.

In Chapter 15, we are interested in approximating $\|A\|_{2\to4}$. We study a natural semidefinite programming (SDP) relaxation for computing the $2 \to 4$ norm of a given linear operator which we call TensorSDP. While TensorSDP is very unlikely to provide a polytime constant-factor approximation for the $2 \to 4$ norm in general (as shown in [34]), we do show that it provides such approximation on random linear operators, as we describe below.

Theorem 3.4.1 (Informal version of Theorem 15.3.1)**.** TensorSDP *certifies a constant upper bound on the ratio $\|A\|_{2\to4}/\,\|A\|_{2\to2}$ where $A : \mathbb{R}^n \to \mathbb{R}^m$ is a* random linear operator *(e.g., obtained by a matrix with entries chosen as i.i.d Bernoulli variables) and $m \geq \Omega(n^2)$.*

In contrast, if $m = o(n^2)$ then this ratio is $\omega(1)$, and hence this result is almost tight in the sense of obtaining "good approximation" in the sense mentionen above.

3.4.2 Hardness of MAXΓ-2-LIN and MAXΓ-3-LIN over integers

To describe our contributions in Chapter 16, we first define the MAXΓ-2-LIN and MAXΓ-3-LIN problems over $\mathbb{Z}$ as follows.

Definition 3.4.2. *In a* MAXΓ-2-LIN *($\mathbb{Z}$) instance, there are n variables $x_1, x_2, \ldots, x_n \in \mathbb{Z}$, and m equations where each equation is in the form $x_i - x_j = c$ where $c \in \mathbb{Z}$. The goal is to find an assignment of the variables such as the fraction of the satisfied equations is maximized.*

In a MAXΓ-3-LIN *($\mathbb{Z}$) instance, there are n variables $x_1, x_2, \ldots, x_n \in \mathbb{Z}$, and m equations where each equation is in the form $x_i + x_j - x_k = c$ where $c \in \mathbb{Z}$. The goal is to find an assignment of the variables such as the fraction of the satisfied equations is maximized.*

Observe that by definition MAXΓ-2-LIN ($\mathbb{Z}$) and MAXΓ-3-LIN ($\mathbb{Z}$) are not ordinary CSPs since the domain size is not constant (even not finite).

In a seminal result by Håstad [116], it was proved that for every constant $\epsilon > 0$, the $(1 - \epsilon, \epsilon)$-gap-MAXΓ-3-LIN ($\mathbb{Z}_q$) problem is NP-hard. Khot et al. [141] proved a similar hardness of approximation result for the MAXΓ-2-LIN ($\mathbb{Z}_q$) problem. They proved that for every constant ϵ, there exists $q > 0$ such that there is no polynomial-time algorithm for the $(1 - \epsilon, \epsilon)$-gap-MAXΓ-2-LIN ($\mathbb{Z}_q$) problem, assuming the Unique Games Conjecture.

Guruswami and Raghanvendra [109] later extended Håstad's theorem to the MAX3-LIN($\mathbb{Z}$) problem, proving that for every constant $\epsilon > 0$, the $(1 - \epsilon, \epsilon)$-gap-MAX3-LIN($\mathbb{Z}$) problem is NP-hard.

We will give a simplified proof of Guruswami and Raghavendra's result, and prove an analogue of the theorem for MAXΓ-3-LIN ($\mathbb{Z}$) assuming the Unique Games Conjecture. In particular, we prove the following theorem.

Theorem 3.4.3. *For every constant $\epsilon > 0$, the $(1 - \epsilon, \epsilon)$-gap-*MAXΓ-3-LIN *($\mathbb{Z}$) problem is* NP*-hard. The proof of this theorem simplifies the proof of the similar result by Guruswami and Raghavendra.*

*Assuming the Unique Games Conjecture, for every constant $\epsilon > 0$, there is no polynomial-time algorithm for the $(1 - \epsilon, \epsilon)$-gap-*MAXΓ-2-LIN *($\mathbb{Z}$) problem.*

3.4.3 Hardness of approximating almost satisfiable MAXHORN3-SAT

Zwick [228] gave an algorithm for the almost satisfiable MAXHORN3-SAT problem. When the input instance is $(1 - \epsilon)$-satisfiable, Zwick's algorithm finds an assignment satisfying a $\left(1 - O\left(\frac{1}{\log(1/\epsilon)}\right)\right)$-fraction of the constraints.

In Chapter 17, we will prove that Zwick's algorithm is essentially optimal assuming the Unique Games Conjecture. In particular, we proved the following theorem.

Theorem 3.4.4 (Pre-statement of Theorem 17.2.1)**.** *Assuming the Unique Games Conjecture[136], For some absolute constant C, for every $\epsilon > 0$, given a $(1 - \epsilon)$-satisfiable instance of* MAXHORN3-SAT*, there is no polynomial-time algorithm to find an assignment satisfying more than a fraction of $\left(1 - \frac{C}{\log(1/\epsilon)}\right)$ of the constraints.*

Part I

Study of the LP/SDP relaxation hierarchies

Chapter 4

Approximation schemes via Sherali-Adams hierarchy for dense constraint satisfaction problems and assignment problems

4.1 Introduction

Recall that in a maximum constraint satisfaction problem (MAX-CSP), given a variable set V over the domain D and a set of constraints $\mathcal{C}$ over the variables in V, we want to find an assignment $\alpha : V \to D$ that maximizes the fraction of constraints satisfied by α. MAX-CSP includes many fundamental problems such as MAX-CUT and MAX-SAT.

In general, MAX-CSP is NP-Hard, and it is even NP-Hard to approximate within a constant factor [20]. However, de la Vega [74] showed that there is a polynomial-time approximation scheme for MAX-CUT if the input graph is *dense*, i.e., it has $\Omega(n^2)$ edges. Here, a *polynomial-time approximation scheme* (PTAS) is an algorithm that, given $\epsilon > 0$ as a parameter, gives a $(1-\epsilon)$-approximation to the optimal value, and runs in polynomial time for any constant ϵ. MAXkCSP is a subproblem of MAX-CSP, in which each constraint involves at most k variables, where k is a constant. Arora et al. [18] and Frieze and Kannan [87] showed PTASs for dense MAXkCSP, i.e., the input instance has $\Omega(n^k)$ constraints. Now it is known that we can compute $(1-\epsilon)$-approximation to the optimal value in time that depends only on k and ϵ [8].

There are two directions to generalize PTASs for dense MAXkCSP. The first one is to generalize the notion of the density condition. We say that an instance of Max-2-CSP is *metric* if the weights of the constraints form a metric. MAX-CUT [77] and MAX-BISECTION [76] admit PTASs if the instance is metric. The notion of *local density* is introduced to generalize the notion of metric to constraints over more than two variables. If the instance is locally dense, MAXkCSP admits PTASs [75].

The second direction is to handle the maximum assignment problems (MAX-AP). In this problem, given a variable set V and a set of constraints, we want to find a permutation π of V to maximize the fraction of satisfied constraints. MAX-AP includes many fundamental problems such as MAXIMUMACYCLICSUBGRAPH, BETWEENNESS, MAXIMUM-GRAPH ISOMORPHISM, DENSEkSUBGRAPH, and QUADRATICASSIGNMENTPROBLEM. MAXkAP is a special case of MAX-AP, in which each constraint involves at most k variables (see Section 4.2 for the precise definition). We say that an instance of MAXkAP is dense if it has $\Omega(n^k)$ constraints. Arora et al. [17] showed a quasi-polynomial-time approximation scheme for dense MAXkAP and PTASs for many special cases.

As we have seen, MAX-CSP and MAXkAP admit PTASs (or quasi-PTASs) in the dense case and the locally dense case. However, the techniques to obtain them vary a lot. For example, [18] is based on the idea of exhaustively trying all assignments for a small number of variables and then solving the rest using the partial assignment. On the other hand, [87] used a variant of Szemerédi's regularity lemma [216]. To deal with the metric case, [77] used the method of copying important variables, and [75] considered a variant of singular value decomposition of tensors to deal with the locally dense case.

4.1.1 Linear Programming (LP) relaxation and LP relaxation hierarchies

Much about LP/SDP relaxations and their hierarchies are introduced at the beginning of this thesis. Here we would like to emphasize that LP relaxation and its hierarchies have found many connections to other known algorithmic frameworks, and to be a unified approach to solve several classes of problems. A few examples are listed as follows. Assuming the Unique Games Conjecture, a canonical LP relaxation (also referred to as the *Basic LP*) is shown to provide optimal approximation guarantee for CSPs with strict constraints [152]. It is known to the author that constant-round Sherali-Adams LP relaxation decides the satisfiability of bounded-width CSPs; Atserias and Maneva [27] recently showed that the Sherali-Adams LP relaxation hierarchy for graph isomorphism interleaves with the levels of pebble-game equivalence with counting (i.e. higher-dimensional Weisfeiler-Lehman color refinement algorithm).

4.1.2 Our contributions

In this chapter, we present the Sherali-Adams LP relaxation hierarchy as a unified approach to dense and locally dense problems – we show that a small number of rounds of the Sherali-Adams LP relaxation gives an approximation scheme to dense MAXkCSP and all their variants studied in the previous works.

Our first main theorem deals with dense and locally dense MAXkCSP.

Theorem 4.1.1 (Informal version of Theorem 4.5.1)**.** *For any $\epsilon > 0$, $O(\frac{1}{\epsilon^2})$-round Sherali-Adams LP relaxation gives $(1-\epsilon)$-approximation to dense or locally dense* MAXkCSP.

Then, we turn to dense and locally dense MAXkCSP with global cardinality constraints. For explanatory purposes, we only consider *bisection* constraint, i.e., the domain is $\{0, 1\}$ and the number of variables that are assigned to 0 should be equal to the number of variables that are assigned to 1. We show that

Theorem 4.1.2 (Informal version of Theorem 4.8.2)**.** *For any $\epsilon > 0$, $O(\frac{1}{\epsilon^2})$-round Sherali-Adams LP relaxation gives $(1-\epsilon)$-approximation to dense or locally dense bisection* MAXkCSP.

Finally, we consider the dense MAXkAP problems, and show that

Theorem 4.1.3 (Informal version of Theorem 4.5.2)**.** *For any $\epsilon > 0$, $O(\frac{\log n}{\epsilon^2})$-round Sherali-Adams LP relaxation gives $(1-\epsilon)$-approximation to dense or locally dense* MAXkAP *problems with n variables.*

In all the precise theorem statements, we actually show additive approximation guarantee (i.e. the value of the rounded solution being at least the fractional optimal value minus a constant error) instead of multiplicative approximation guarantee. However, since we define the problems in a way that the optimal solution is $\Omega(1)$ (see Section 4.2 for the precise definition of the problems), an additive approximation scheme implies a multiplicative approximation scheme.

New algorithmic guarantees. Let us define the problem MAXIMUMk-HYPERGRAPHISOMORPHISM as follows. Given two weighted k-uniform hypergraph $G = (V, \omega')$ and $H = (V, \omega'')$, where $\omega', \omega'' : V^k \to [0, 1]$ are the weight functions over all possible hyperedges. The goal is to find a permutation π over V so that $\sum_{e \in V^k} \omega'(e)\omega''(\pi(e))$ is maximized (where $\pi(e)$ is the edge obtained by applying π on each incident vertex of e). It is easy to see Theorem 4.1.3 implies that $O(\frac{\log n}{\epsilon^2})$-round

Sherali-Adams LP relaxation gives $(1 - \epsilon)$-approximation to MAXIMUMk-HYPERGRAPHISOMORPHISM when both G and H are dense.

We are able to apply our analysis framework for the Sherali-Adams LP relaxation to another special case of MAXIMUMk-HYPERGRAPHISOMORPHISM, getting the following new algorithmic guarantee.

Theorem 4.1.4 (Informal version of Theorem 4.5.3)**.** *For any $\epsilon > 0$, $O(\frac{\log n}{\epsilon^2})$-round Sherali-Adams LP relaxation gives $(1 - \epsilon)$ approximation to the* MAXIMUM k-HYPERGRAPHISOMORPHISM *problem when one of the two graphs is locally dense and the other graph is dense, where n is the number of vertices in the hypergraphs. Therefore, this special case of the problem admits a $(1 - \epsilon)$-approximation algorithm in time $n^{O(\frac{\log n}{\epsilon^2})}$.*

4.1.3 Proof overview

The first step of our algorithms is to condition on a set of random variables in a solution to the Sherali-Adams LP relaxation. In the *ℓ-round Sherali-Adams LP relaxation* (or the SA relaxation for short), for each set of variables S of size at most ℓ, we have a probability distribution $\boldsymbol{\mu}_S$ over assignments on S. First we solve $(k+\ell)$-round SA relaxation, where ℓ is a parameter depending on the error parameter ϵ. Then, we randomly sample a set of variables $u_1, \ldots, u_\ell$ and assign values to them by sampling values from $\boldsymbol{\mu}_{\{u_1\}}, \ldots, \boldsymbol{\mu}_{\{u_\ell\}}$, respectively. By this conditioning, we obtain a solution to k-round Sherali-Adams relaxation $\boldsymbol{\mu}'$ with the same LP value in expectation. An important fact here is that variables become almost independent in the sense that, if we sample a k-tuple $(v_1, \ldots, v_k)$ according to a dense (or locally dense) distribution (this distribution corresponds to the weights of the constraints in kCSPand kAP instances), the distribution $\boldsymbol{\mu}_{\{v_1,\ldots,v_k\}}$ and the product distribution $\boldsymbol{\mu}_{\{v_1\}} \times \cdots \times \boldsymbol{\mu}_{\{v_k\}}$ are close in expectation.

The second step of our algorithms is to round the solution to the SA relaxation where the variables are almost independent. For dense (or locally dense) kCSPand bisection kCSP, the rounding algorithm just samples a value from $\boldsymbol{\mu}_{\{v\}}$ and assigning it to v for each variable v. It is relatively easy to show that the expected value of the sampled solution is close to the LP value, and therefore gives a $(1 - \epsilon)$-approximation.

For kAP problems, however, such independent sampling method does not work – there might be more than one variables assigned to the same value and we do not get a permutation when this happens. Instead, we view the marginal probability distributions on single variables, $\boldsymbol{\mu}_{\{u\}}(w)$, as a doubly stochastic matrix. We view this doubly stochastic matrix as a probability distribution of permutations. We iteratively choose two permutations in the support of the distribution and merge them into a new permutation, until there is only one permutation left in the support – which is the output of our rounding algorithm. The operation of merging two permutations is interestingly similar to the merging operation used in [17], although the purposes are different. See Section 4.4.2 for more details.

4.1.4 Comparison to previous works

We first compare the running time of our SA relaxation-based algorithms with the previously known counterparts. For MAXkCSP, the running time $n^{O(1/\epsilon^2)}$ of our method matches the one of the method by [18]. For MAXkAP the running time $n^{O(\log n/\epsilon^2)}$ of our method matches the one of the method by [17]. [17] improved the running time to $n^{O(1/\epsilon^2)}$ for various problems by reducing them to CSPs. We can use the same techniques to obtain the same running time for these problems.

The number of rounds ($O(\frac{1}{\epsilon^2})$) in Theorem 4.1.1 improves the corresponding theorem in [78] which showed that $\tilde{O}(\frac{1}{\epsilon^4})$-round SA relaxation gives $(1-\epsilon)$-approximation to dense MAX-CUT.

The idea of conditioning variables of a solution to LP/SDP hierarchies is used in [195, 36] to solve variants of MAX2CSP. Let $G = (V, E)$ be the underlying graph of an instance of MAX2CSP. Barak et al. [36] showed that (i) the covariance between u and v over V^2 gets close to zero by conditioning, and (ii) the covariance between u and v over E gets close to the covariance between u and v over V^2 by conditioning if G is expander-like. Combining these two results, they show a PTAS for MAX2CSP when G is expander-like. This method can be also applied to dense graphs, but it is not clear how to generalize it to metric graphs and kCSP.

Raghavendra and Tan [195] used mutual information instead of covariance to measure correlation between two variables and simplified the proof. They noticed that conditioning is useful to deal with global constraints such as cardinality constraints since after conditioning we can sample variables independently and the resulting solution will not break global constraints much. With this idea, they gave a 0.85-approximation algorithm for MAX-BISECTION. Though our method and analysis are similar to theirs, we use the independence for obtaining PTASs for the dense and locally dense case as well as supporting global constraints. Also, to handle constraints of larger arities, we use total correlation instead of mutual information to measure correlation among variables.

Coja-Oghlan et al. [70] showed that, even if the instance is sparse, if it satisfies a certain pseudo-random condition, then MAXkCSP admits PTASs. If $k = 2$, this results can be seen as a special case of [195] because the pseudo-random condition would imply that the underlying graph is expander-like. Their result is incomparable to ours because it is not clear how the pseudo-random condition and the locally dense condition.

4.1.5 Organization

In Section 4.2, we introduce definitions and notions used in this paper. In Section 4.3, we show an algorithm that obtains an almost independent solution to the Sherali-Adams LP relaxation. Section 4.4 is devoted to show how to round the obtained solution to the Sherali-Adams LP relaxation. We combine the two steps together in Section 4.5. Section 4.6 and Section 4.7 are devoted to prove auxiliary lemmas. We consider CSPs with global cardinality constraints in Section 4.8.

4.2 Preliminaries

For an integer $a \geq 1$, $[a]$ denotes the set $\{1, \ldots, a\}$. For a set Y and $0 \leq k \leq |Y|$, $\binom{Y}{k}$ denotes the family of sets $X \subseteq Y$ with $|X| = k$. We usually use V to denote the set of variables in a problem, and use $n = |V|$ to denote the number of variables. For an event A, $\mathbf{1}[A]$ denotes the corresponding indicator function.

Probability theoretic notions: We recall several notions from probability theory. For a probability distribution $\boldsymbol{\mu}$ on Ω, $\mathrm{supp}(\boldsymbol{\mu})$ denotes the *support* of $\boldsymbol{\mu}$, i.e., $\mathrm{supp}(\boldsymbol{\mu}) = \{i \in \Omega \mid \boldsymbol{\mu}(i) > 0\}$. For a set S, $i \sim S$ means that we sample i uniformly at random from S.

Let $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ be two probability distributions on a finite set Ω. Then, the L_1 distance between them is defined as $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_1 = \sum_{i \in \Omega} \|\boldsymbol{\mu}_1(i) - \boldsymbol{\mu}_2(i)\|$. The Kullback-Leibler divergence between them is defined as $d_{KL}(\boldsymbol{\mu}_1 \| \boldsymbol{\mu}_2) = \sum_{i \in \Omega} \boldsymbol{\mu}_1(i) \log \frac{\boldsymbol{\mu}_1(i)}{\boldsymbol{\mu}_2(i)}$. and the Kullback-Leibler divergence $d_{KL}(\boldsymbol{\mu}_1 \| \boldsymbol{\mu}_2)$ are defined as follows. We provide the following fact without proof.

Lemma 4.2.1. *Let $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ be two probability distributions on a finite set Ω. Then,* $\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_1 \leq \sqrt{2 d_{KL}(\boldsymbol{\mu}_1 \| \boldsymbol{\mu}_2)}$.

Information theoretic notions: We now recall some definitions from information theory. For a random variable $\boldsymbol{x}$, $\boldsymbol{\mu}_{\boldsymbol{x}}$ denotes the corresponding probability distribution. That is, for any i, we have $\boldsymbol{\mu}_{\boldsymbol{x}}(i) = \Pr[\boldsymbol{x} = i]$.

Let $\boldsymbol{x}$ be a random variable on a finite set Ω. The *entropy* of $\boldsymbol{x}$ is defined as $H(\boldsymbol{x}) = -\sum_{i\in\Omega} \Pr[\boldsymbol{x} = i] \log \Pr[\boldsymbol{x} = i]$.

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be jointly distributed variables on a finite set Ω. The *entropy* of $\boldsymbol{x}$ conditioned on $\boldsymbol{y}$ is defined as $H(\boldsymbol{x} \mid \boldsymbol{y}) = \mathop{\mathbf{E}}_{i\sim\boldsymbol{\mu}_{\boldsymbol{y}}}[H(\boldsymbol{x} \mid \boldsymbol{y} = i)]$. The *mutual information* of $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as $I(\boldsymbol{x};\boldsymbol{y}) = d_{KL}(\boldsymbol{\mu}_{(\boldsymbol{x},\boldsymbol{y})} \| \boldsymbol{\mu}_{\boldsymbol{x}} \times \boldsymbol{\mu}_{\boldsymbol{y}})$.

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ $(k \geq 2)$ be jointly distributed variables on a finite set Ω. The *mutual information* of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ is defined as $I(\boldsymbol{x}_1; \ldots; \boldsymbol{x}_k) = I(\boldsymbol{x}_1; \ldots; \boldsymbol{x}_{k-1}) - I(\boldsymbol{x}_1; \ldots; \boldsymbol{x}_{k-1} \mid \boldsymbol{x}_k)$, where $I(\boldsymbol{x}_1; \ldots; \boldsymbol{x}_{k-1} \mid \boldsymbol{x}_k) = \mathbf{E}_{i\sim\boldsymbol{\mu}_{\boldsymbol{x}_k}}[I(\boldsymbol{x}_1; \ldots; \boldsymbol{x}_{k-1} \mid \boldsymbol{x}_k = i)]$. The *total correlation* of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ is defined as $C(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) = d_{KL}(\boldsymbol{\mu}_{(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_k)} \| \boldsymbol{\mu}_{\boldsymbol{x}_1} \times \cdots \times \boldsymbol{\mu}_{\boldsymbol{x}_k})$.

We give two well-known facts in information theory below.

Lemma 4.2.2. *Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two jointly distributed variables on a finite set Ω. Then*

$$I(\boldsymbol{x};\boldsymbol{y}) = H(\boldsymbol{x}) - H(\boldsymbol{x} \mid \boldsymbol{y}).$$

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ be jointly distributed variables on a finite set Ω. Then

$$I(\boldsymbol{x}_1; \ldots; \boldsymbol{x}_k) = \sum_{(i_1,\ldots,i_t)\subseteq[k], t\geq 1} (-1)^{t-1} H(\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_t}).$$

Lemma 4.2.3. *Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ be jointly distributed variables on a finite set Ω. Then*

$$C(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) = \sum_{(i_1,\ldots,i_t)\subseteq[k], t\geq 2} I(\boldsymbol{x}_{i_1}; \ldots; \boldsymbol{x}_{i_t}).$$

Constraint satisfaction problems: Let D be a nonempty finite domain and $k \geq 2$ be an integer. An *instance* $\mathcal{I} = (V, \omega, P)$ of kCSPconsists of a set V of *variables*, a *scope distribution* ω over V^k, and a set of *payoff functions* $P = \{P_S : D^S \to [0,1] \mid S \in V^k\}$. An *assignment* for an instance $\mathcal{I} = (V, \omega, P)$ is a mapping $\alpha : V \to D$. The *value* of the assignment, denoted $\mathsf{val}(\mathcal{I}, \alpha) \in [0,1]$, is defined as $\mathsf{val}(\mathcal{I}, \alpha) = \Pr_{S\sim\omega}[P_S(\alpha|_S)]$, where $\alpha|_S$ is the projection of α to S. We define the *optimum value* of the instance $\mathcal{I}$ to be $\mathsf{opt}(\mathcal{I}) = \max_\alpha\{\mathsf{val}(\mathcal{I}, \alpha)\}$.

Let $\mathcal{I} = (V, \omega, P)$ be an instance of CSP. A *solution* to the ℓ-round Sherali-Adams relaxation consists of a probability distribution $\boldsymbol{\mu}_S$ over D^S for each set $S \subseteq V$ of size at most ℓ. The objective function is the probability that α is in P_S, where S is sampled from ω and α is sampled from $\boldsymbol{\mu}_S$. Strictly speaking, we sample a tuple $(v_1, \ldots, v_k)$ from ω, but we regard it as the set $\{v_1, \ldots, v_k\}$ when we use it as a subscript of $\boldsymbol{\mu}$. In other words, $\boldsymbol{\mu}_S$

and $\boldsymbol{\mu}_T$ are the same distribution for two tuples S and T if they are the same as sets. Also, for every pair of sets S and T with $|S \cup T| \leq \ell$, the corresponding probability distributions $\boldsymbol{\mu}_S$ and $\boldsymbol{\mu}_T$ must be consistent on $S \cap T$. Formally, the ℓ-round Sherali-Adams relaxation for a kCSPinstance $\mathcal{I} = (V, \omega, P)$ ($\ell \geq k$) is written as follows.

$$\begin{array}{ll} \text{maximize} & \mathop{\mathbf{E}}_{S\sim\omega} \mathop{\mathbf{E}}_{\alpha\sim\boldsymbol{\mu}_S}[P_S(\alpha)] \\ \text{subject to} & \Pr_{\alpha\sim\boldsymbol{\mu}_S}[\alpha|_{S\cap T} = \beta] = \Pr_{\alpha\sim\boldsymbol{\mu}_T}[\alpha|_{S\cap T} = \beta] \quad \forall S, T \subseteq V, |S \cup T| \leq \ell, \beta \in D^{S\cap T}. \end{array}$$

It is not hard to see that the relaxation above can be written as a linear programming (see, e.g., [191] for details). We define $\boldsymbol{x}_v$ as the random variable sampled from the distribution $\boldsymbol{\mu}_{\{v\}}$. We use $\mathsf{val}_{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu})$ to denote the objective value of the LP solution $\boldsymbol{\mu}$. The same definition applies to the following subsections.

Assignment problems: The assignment problem differs from CSP in that we want to maximize the objective function over the set of permutations. Similarly to CSP, for an integer $k \geq 2$, an instance of the degree-k assignment problem is given as $\mathcal{I} = (V, \omega)$, where V is the set of variables, ω is a distribution over $V^k \times V^k$. The *scope distribution* of $\mathcal{I}$ is the marginal distribution of ω on the first k elements. An *assignment* for an instance $\mathcal{I} = (V, \omega)$ is a permutation π of V. The *value* of the assignment π, denoted $\mathsf{val}(\mathcal{I}, \pi)$, is defined as

$$\mathsf{val}(\mathcal{I}, \pi) = n^k \Pr_{(U,W)\sim\omega}[\forall i \in [k] : \pi(u_i) = w_i],$$

where $U = (u_1, u_2, \ldots, u_k)$ and $W = (w_1, w_2, \ldots, w_k)$. We define the optimum value of $\mathcal{I}$ to be $\mathsf{opt}(\mathcal{I}) = \max_\pi\{\mathsf{val}(\mathcal{I}, \pi)\}$.

Though the definition of $\mathsf{val}(\mathcal{I}, \pi)$ may look non-standard, it is just the objective function used in [17] with a normalization factor that is multiplied to make the optimum $\Omega(1)$ when ω is dense.

The ℓ-round Sherali-Adams relaxation of an kAP instance $\mathcal{I} = (V, \omega)$ ($\ell \geq k$) is as follows.

$$\begin{array}{ll} \text{maximize} & \mathop{\mathbf{E}}_{(U,W)\sim\omega} \Pr_{\alpha\sim\boldsymbol{\mu}_U}[\forall i \in [k] : \alpha(u_i) = w_i] \\ \text{subject to} & \Pr_{\beta\sim\boldsymbol{\mu}_S}[\beta|_{S\cap T} = \alpha] = \Pr_{\beta\sim\boldsymbol{\mu}_T}[\beta|_{S\cap T} = \alpha] \qquad \forall S, T \subseteq V, |S \cup T| \leq \ell, \alpha \in V^{S\cap T} \\ & \displaystyle\sum_{\alpha\in V^S} \sum_{u\in V\setminus S} \boldsymbol{\mu}_{S\cup\{u\}}(\alpha \cup \{u \to w\}) = \sum_{\alpha\in V^S} \boldsymbol{\mu}_S(\alpha) \quad \forall w \in V, S \subseteq V, |S| < \ell. \end{array}$$

The difference from the Sherali-Adams relaxation for CSP is that we have extra constraints in the last line whose intended meaning is that each value i can be taken by at most one variable.

Density condition: We now introduce the notion of dense and locally dense distributions.

Let ω be a probability distribution over a finite set Ω. For $\Delta \in (0,1]$, we say ω is Δ-*dense* if for every $a \in \Omega$, it holds that $\Delta \cdot \omega(a) \leq \frac{1}{|\Omega|}$.

Let ω be a probability distribution over V^k. Let $d_i(v) = \Pr_{S \sim \omega}[S_i = v_i]$ be the probability that the i-th coordinate is v under ω. For $\Delta \in (0,1]$, we say ω is Δ-*locally dense* if for every $(v_1, \dots, v_k) \in V^k$, it holds that

$$\Delta \cdot \omega(v_1, \dots, v_k) \leq \frac{1}{|V|^{k-1}} \sum_{1 \leq i \leq k} d_i(v_i).$$

Since $d_i(v) = \sum_{S \in V^k : S_i = v} \omega(S)$, the RHS of the locally dense condition is equal to $\sum_{1 \leq i \leq k} \mathop{\mathbf{E}}_{S \sim V^k}[\omega(S) \mid S_i = v_i]$. Thus the locally dense condition says that no tuple $(v_1, \dots, v_k)$ is "wild" in that $\omega(v_1, \dots, v_k)$ is at most constant times the sum over i of the average probability mass of S with $S_i = v_i$.

The notion of local density is introduced in [75] to generalize the metric condition. To see this, suppose $\omega : V^2 \to \mathbb{R}$ is a metric. Then, ω is 1-locally dense since, for any $u, v \in V$, we have $\frac{1}{n}(d_1(u) + d_2(v)) = \frac{1}{n}\sum_w (\omega(u,w) + \omega(w,v)) \geq \frac{1}{n}\sum_w \omega(u,v) \geq \omega(u,v)$.

It is immediate to verify the following lemma.

Lemma 4.2.4. *Let ω be a probability distribution over $\Omega_1 \times \Omega_2$. If ω is Δ-dense (resp., Δ-locally dense), then the marginal distribution ω_1 of ω on Ω_1 is also Δ-dense (resp., Δ-locally dense).*

4.3 Conditioning operations for Sherali-Adams LP hierarchy

Recall that, a solution to the ℓ-round SA relaxation consists of distributions over sets of ℓ variables. In this section, we show that, if the scope distribution is dense or locally dense, then by conditioning a small number of variables, we can make variables almost independent in these distributions. Once variables become almost independent, we can round variables independently without losing the objective value much (see Section 4.4).

Let $\mathcal{I}$ be an instance of kCSPor kAP with a variable set V. Fix ℓ and let $\boldsymbol{\mu}$ be a solution to the ℓ-round Sherali-Adams relaxation. For a variable set $S = (v_1, \dots, v_k)$,

$C_{\boldsymbol{\mu}}(\boldsymbol{x}_S)$ denotes the total correlation $C(\boldsymbol{x}_{v_1}, \ldots, \boldsymbol{x}_{v_k})$ under the probability distribution $\boldsymbol{\mu}_S$. We use the following notion to measure independence of variables.

Definition 4.3.1. *Let $\mathcal{I}$ be an instance of kCSPor kAP with a scope distribution ω. A solution $\boldsymbol{\mu}$ to the ℓ-round SA relaxation for $\mathcal{I}$ with $\ell \geq k$ is κ-*independent with respect to distribution *ω' if*

$$\mathop{\mathbf{E}}_{S \sim \omega'}[C_{\boldsymbol{\mu}}(\boldsymbol{x}_S)] \leq \kappa.$$

*We say that $\boldsymbol{\mu}$ is κ-*independent *if it is κ-independent with respect to ω.*

In Section 4.3.1, we explain how to condition variables. In Section 4.3.2 and Section 4.3.3, we show that the conditioning operation outputs κ-independent LP solutions for the dense case and the locally dense case, respectively.

4.3.1 Conditioning operations

We first describe the operation of conditioning one variable. Given a solution $\boldsymbol{\mu}$ to the ℓ-round SA relaxation with $\ell \geq 2$, we sample a vertex u uniformly at random and then set $\boldsymbol{x}_u = i$, where i is a value sampled from $\boldsymbol{\mu}_{\{u\}}$. This operation gives a solution $\boldsymbol{\mu}'$ to the $(\ell-1)$-round SA relaxation: For each tuple $(v_1, \ldots, v_{\ell-1})$ of $\ell - 1$ variables, we define $\boldsymbol{\mu}'_{\{v_1,\ldots,v_{k-1}\}}(i_1, i_2, \ldots, i_{k-1}) = \boldsymbol{\mu}_{\{v_1,\ldots,v_{k-1},u\}}(i_1, i_2, \ldots, i_{k-1}, i)$. It is not hard to check that $\boldsymbol{\mu}'$ is indeed a solution to the $(\ell-1)$-round SA relaxation.

Our algorithm is given in Algorithm 1. Given an solution $\boldsymbol{\mu}$ to the $(\ell + \ell')$-round SA relaxation, it iteratively conditions variables. We will show in subsequent sections that, if ω is Δ-dense or Δ-locally dense, then Algorithm 1 outputs a κ-independent LP solution in ℓ' steps on average, where $\kappa = \frac{k4^k \log |D|}{\ell'}$. (If ω is Δ-dense, κ can be slightly smaller.)

We mention here the following simple fact.

Lemma 4.3.2. *Let $\boldsymbol{\mu}'$ be the solution output by Algorithm 1. Then,* $\mathbf{E}\,\mathsf{val}^{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}') = \mathsf{val}^{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu})$.

Proof. Notice that the algorithm respects the marginal distributions provided by the SA relaxation during sampling the values to variables. Thus, the expected objective value is preserved. □

Algorithm 1 Conditioning operation of Sherali-Adams solutions

Input: A feasible solution $\boldsymbol{\mu}$ to the $(\ell+\ell')$-round SA relaxation for a CSP instance $\mathcal{I} = (V,\omega)$.

Output: An κ-independent solution to the ℓ-round SA relaxation, where $\kappa = \frac{k4^k \log|D|}{\ell'}$

Set $t = 1$.

while the current LP solution is not κ-independent **do**

Sample a variable $u_t \in V$ uniformly at random.

Sample a value a from its marginal distribution $\boldsymbol{\mu}_{\{u_t\}}$ after the first $t-1$ fixings, and condition the LP solution by setting $\boldsymbol{x}_{u_t} = a$.

$t = t+1$.

4.3.2 The dense case

We consider the dense case, that is, ω is a uniform distribution.

Lemma 4.3.3. *If ω is uniform distribution over V^k, there exists $t \le \ell'$ such that*

$$\mathop{\mathbf{E}}_{U\sim V^t}\mathop{\mathbf{E}}_{S\sim V^k}[C_{\boldsymbol{\mu}}(\boldsymbol{x}_S \mid \boldsymbol{x}_U)] \le \frac{3^k \log|D|}{\ell'}.$$

Proof. We consider the value

$$\sum_{1\le t\le \ell'}\mathop{\mathbf{E}}_{U\sim V^t}\mathop{\mathbf{E}}_{S\sim V^k}[C_{\boldsymbol{\mu}}(\boldsymbol{x}_S \mid \boldsymbol{x}_U)].$$

From Lemma 4.2.3, this value can be decomposed as

$$\sum_{1\le t\le \ell'}\mathop{\mathbf{E}}_{U\sim V^t}\mathop{\mathbf{E}}_{S\sim V^k}\left[\sum_{2\le r\le k}\sum_{R\in\binom{S}{r}} I_{\boldsymbol{\mu}}(\boldsymbol{x}_R \mid \boldsymbol{x}_U)\right] = \sum_{2\le r\le k}\binom{k}{r}\sum_{1\le t\le \ell'}\mathop{\mathbf{E}}_{U\sim V^t}\mathop{\mathbf{E}}_{R\sim V^r}\left[I_{\boldsymbol{\mu}}(\boldsymbol{x}_R \mid \boldsymbol{x}_U)\right],$$

where for a set $R = (v_1,\ldots,v_r)$, $I_{\boldsymbol{\mu}}(\boldsymbol{x}_R)$ denotes the mutual information $I_{\boldsymbol{\mu}_R}(v_1;\ldots;v_r)$.

To bound this value, we recall the definition of mutual information. For any $t \le \ell'$,

$$\mathop{\mathbf{E}}_{\substack{U\sim V^t\\ R\sim V^r}}[I_{\boldsymbol{\mu}}(\boldsymbol{x}_R \mid \boldsymbol{x}_U)] = \mathop{\mathbf{E}}_{\substack{U\sim V^t\\ R\sim V^{r-1}}}[I_{\boldsymbol{\mu}}(\boldsymbol{x}_R \mid \boldsymbol{x}_U)] - \mathop{\mathbf{E}}_{\substack{U\sim V^{t+1}\\ R\sim V^{r-1}}}[I_{\boldsymbol{\mu}}(\boldsymbol{x}_R \mid \boldsymbol{x}_U)].$$

Adding the equalities from $t = 0$ to ℓ', we get

$$\sum_{0 \le t \le \ell'} \mathop{\mathbf{E}}_{\substack{U \sim V^t \\ R \sim V^r}} [I_{\boldsymbol{\mu}}(\boldsymbol{x}_R \mid \boldsymbol{x}_U)] = \mathop{\mathbf{E}}_{R \sim V^{r-1}} [I(\boldsymbol{x}_R)] - \mathop{\mathbf{E}}_{\substack{U \sim V^{\ell'+1} \\ R \sim V^{r-1}}} [I_{\boldsymbol{\mu}}(\boldsymbol{x}_R \mid \boldsymbol{x}_U)] \le 2^r \log |D|,$$

where the last inequality holds from $I_{\boldsymbol{\mu}}(\boldsymbol{x}_R) \le 2^{|R|} \log |D|$ by Lemma 4.2.2. Thus, we have

$$\sum_{0 \le t \le \ell'} \mathop{\mathbf{E}}_{U \sim V^t} \mathop{\mathbf{E}}_{S \sim V^k} [C(\boldsymbol{x}_S \mid \boldsymbol{x}_U)] \le 3^k \log |D|,$$

and the lemma follows. □

The following corollary is immediate.

Corollary 4.3.4. *If ω is a Δ-dense distribution over V^k. Then there exists $t \le \ell'$ such that*

$$\mathop{\mathbf{E}}_{U \sim V^t} \mathop{\mathbf{E}}_{S \sim \omega} [C_{\boldsymbol{\mu}}(\boldsymbol{x}_S \mid \boldsymbol{x}_U)] \le \frac{3^k \log |D|}{\Delta \ell'}.$$

4.3.3 The locally dense case

We now consider the case that the scope distribution ω is 1-locally dense.

Lemma 4.3.5. *If ω is a 1-locally dense distribution over V^k, then there exists $t \le \ell'$ such that*

$$\mathop{\mathbf{E}}_{U \sim V^t} \mathop{\mathbf{E}}_{S \sim \omega} [C_{\boldsymbol{\mu}}(\boldsymbol{x}_S \mid \boldsymbol{x}_U)] \le \frac{k 4^k \log k|D|}{\ell'}.$$

Proof. We consider the value

$$\sum_{1 \le t \le \ell'} \mathop{\mathbf{E}}_{U \sim V^t} \mathop{\mathbf{E}}_{S \sim \omega} [C_{\boldsymbol{\mu}}(\boldsymbol{x}_S \mid \boldsymbol{x}_U)]. \tag{4.1}$$

From Lemma 4.2.3, this value can be decomposed as

$$\sum_{1 \le t \le \ell'} \mathop{\mathbf{E}}_{U \sim V^t} \mathop{\mathbf{E}}_{S \sim \omega} \left[\sum_{2 \le r \le k} \sum_{R \in \binom{S}{r}} I_{\boldsymbol{\mu}}(\boldsymbol{x}_R \mid \boldsymbol{x}_U) \right]$$
$$= \sum_{J \subseteq [k] : 2 \le |J| \le k} \sum_{1 \le t \le \ell'} \mathop{\mathbf{E}}_{U \sim V^t} \mathop{\mathbf{E}}_{R \sim \omega|_J} [I_{\boldsymbol{\mu}}(\boldsymbol{x}_R \mid \boldsymbol{x}_U)],$$

where $\omega|_J$ denotes the marginal distribution of ω on J.

Fix $J \subseteq [k]$ with $|J| = r \geq 2$. Let ω_i be the marginal distribution of ω on the i-th coordinate. Let $\Omega_i = \omega_i \times V^{r-1}$ and $\Omega_i' = \omega_i \times V^{r-2}$. We first analyze $I(\boldsymbol{x}_R)$ under Ω_i instead of $\omega|_J$.

From the definition, for any i and $t \leq \ell'$,

$$\mathop{\mathbf{E}}_{\substack{U\sim V^t\\ R\sim\Omega_i}} I(\boldsymbol{x}_R \mid \boldsymbol{x}_U) = \mathop{\mathbf{E}}_{\substack{U\sim V^t\\ R\sim\Omega_i'}} I(\boldsymbol{x}_R \mid \boldsymbol{x}_U) - \mathop{\mathbf{E}}_{\substack{U\sim V^{t+1}\\ R\sim\Omega_i'}} I(\boldsymbol{x}_R \mid \boldsymbol{x}_U).$$

Adding the equalities from $t = 0$ to $t = \ell'$, we get

$$\sum_{0\leq t\leq \ell'} \mathop{\mathbf{E}}_{\substack{U\sim V^t\\ R\sim\Omega_i}} I(\boldsymbol{x}_R \mid \boldsymbol{x}_U) = \mathop{\mathbf{E}}_{R\sim\Omega_i'} I(\boldsymbol{x}_R) - \mathop{\mathbf{E}}_{\substack{U\sim V^{\ell'+1}\\ R\sim\Omega_i'}} H(\boldsymbol{x}_R \mid \boldsymbol{x}_U) \leq 2^r \log|D|. \tag{4.2}$$

Now we turn to analyze $I(\boldsymbol{x}_{v_1}; \cdots; \boldsymbol{x}_{v_r})$ under $\omega|_J$.

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\substack{U\sim V^t\\ R\sim\omega|_J}} I(\boldsymbol{x}_R \mid \boldsymbol{x}_U) = \mathop{\mathbf{E}}_{U\sim V^t} \sum_{R\sim V^r} \omega|_J(R) I(\boldsymbol{x}_R \mid \boldsymbol{x}_U) \\
&\leq \mathop{\mathbf{E}}_{U\sim V^t} \sum_{(v_1,\ldots,v_r)\sim V^r} \frac{1}{n^{r-1}} \sum_{1\leq i\leq r} d_i(v_i) I(\boldsymbol{x}_{v_1}; \cdots; \boldsymbol{x}_{v_r} \mid \boldsymbol{x}_U) \\
&\hspace{16em} \text{(by local density and Lemma 4.2.4)} \\
&= \sum_{1\leq i\leq r} \mathop{\mathbf{E}}_{U\sim V^t} \sum_{(v_1,\ldots,v_r)\sim V^r} \frac{1}{n^{r-1}} \Pr_{S\sim\omega}[S_i = v_i] I(\boldsymbol{x}_{v_1}; \cdots; \boldsymbol{x}_{v_r} \mid \boldsymbol{x}_U) \\
&= \sum_{1\leq i\leq r} \mathop{\mathbf{E}}_{U\sim V^t} \sum_{R\sim V^r} \Omega_i(R) I(\boldsymbol{x}_{v_1}; \cdots; \boldsymbol{x}_{v_r} \mid \boldsymbol{x}_U) \\
&= \sum_{1\leq i\leq r} \mathop{\mathbf{E}}_{\substack{U\sim V^t\\ R\sim\Omega_i}} I(\boldsymbol{x}_R \mid \boldsymbol{x}_U).
\end{aligned}$$

Thus from (4.2),

$$\sum_{0\leq t\leq \ell'} \mathop{\mathbf{E}}_{\substack{U\sim V^t\\ R\sim\omega|_J}} I(\boldsymbol{x}_R \mid \boldsymbol{x}_U) \leq \sum_{1\leq i\leq r} 2^r \log|D| = r2^r \log|D|.$$

It follows that (4.1) $\leq k4^k \log|D|$ and the lemma holds. $\square$

The following corollary is immediate.

Corollary 4.3.6. *If ω is a Δ-locally dense distribution over V^k, then there exists $t \le \ell'$ such that*

$$\mathop{\mathbf{E}}_{U \sim V^t} \mathop{\mathbf{E}}_{S \sim \omega} [C_{\boldsymbol{\mu}}(\boldsymbol{x}_S \mid \boldsymbol{x}_U)] \le \frac{k 4^k \log |D|}{\Delta \ell'}.$$

4.4 Rounding κ-independent solutions

4.4.1 Constraint satisfaction problems

Lemma 4.4.1. *Let $\mathcal{I} = (V, \omega, P)$ be a kCSPinstance over finite domain D. Let $\boldsymbol{\mu}$ be a κ-independent solution to the k-round Sherali-Adams LP relaxation. There is a randomized polynomial time algorithm to find an assignment $\alpha : V \to D$ such that $\mathsf{val}(\mathcal{I}, \alpha) \ge \mathsf{val}^{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}) - 3\sqrt{\kappa}$.*

Proof. For each $v \in V$, let $\alpha(v)$ be independently sampled from $\boldsymbol{\mu}_{\{v\}}$. For each $S \in V^k$, by the definition of total correlation, Lemma 4.2.1, and the fact that $P_S(\beta) \in [0,1]$ we have

$$\left| \mathop{\mathbf{E}}_{\beta \sim \boldsymbol{\mu}_S} P_S(\beta) - \mathop{\mathbf{E}}_{\alpha} P_S(\alpha|_S) \right| \le 2\sqrt{C(\boldsymbol{x}_S)}.$$

Therefore by κ-independence,

$$\left| \mathop{\mathbf{E}}_{S \sim \omega} \left(\mathop{\mathbf{E}}_{\beta \sim \boldsymbol{\mu}_S} P_S(\beta) - \mathop{\mathbf{E}}_{\alpha} P_S(\alpha|_S) \right) \right| \le \mathop{\mathbf{E}}_{S \sim \omega} 2\sqrt{C(\boldsymbol{x}_S)} \le 2\sqrt{\mathop{\mathbf{E}}_{S \sim \omega} C(\boldsymbol{x}_S)} \le 2\sqrt{\kappa}\ .$$

We have proved that $\mathbf{E}_\alpha[\mathsf{val}(\mathcal{I}, \alpha)] \ge \mathsf{val}^{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}) - 2\sqrt{\kappa}$. Therefore we can sample an α in expected polynomial time such that $\mathsf{val}(\mathcal{I}, \alpha) \ge \mathsf{val}^{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}) - 3\sqrt{\kappa}$. □

4.4.2 Assignment problems

Let $\mathcal{I} = (V, \omega)$ be a Δ-dense kAP instance. We introduce the following relaxation $\mathcal{H}$, and let $\mathsf{val}_{\mathcal{H}}(\mathcal{I})$ be its optimal value.

$$\begin{array}{lll} \text{maximize} & n^k \mathop{\mathbf{E}}_{(U,W)\sim\omega} \prod_{i=1}^{k} \boldsymbol{x}_{u_i,w_i} & \\ \text{subject to} & \boldsymbol{x}_{u,w} \geq 0 & \forall u, w \in V^k \\ & \sum_{u\in V} \boldsymbol{x}_{u,w} = 1 & \forall w \in V \\ & \sum_{w\in V} \boldsymbol{x}_{u,w} = 1 & \forall u \in V. \end{array}$$

4.4.2.1 From κ-independence to relaxation $\mathcal{H}$

We first see that we can find a good solution to $\mathcal{H}$ using a solution to the Sherali-Adams LP relaxation of a dense instance $\mathcal{I}$.

Lemma 4.4.2. *Let $\mathcal{I} = (V, \omega)$ be a kAP instance such that ω is Δ-dense. Let $\boldsymbol{\mu}$ be a κ-independent solution (with respect to the uniform distribution rather than ω) to the k-round Sherali-Adams LP relaxation of $\mathcal{I}$. There is a polynomial-time algorithm, on input $\boldsymbol{\mu}$, to find a solution to $\mathcal{H}$ that certifies that $\mathsf{val}_{\mathcal{H}}(\mathcal{I}) \geq \mathsf{val}_{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}) - 2\sqrt{\kappa}/\Delta$.*

Proof. Let $\boldsymbol{x}_{u,w} = \boldsymbol{\mu}_u(w)$ for all $u, w \in V$. For each $S = (u_1, u_2, \ldots, u_k) \in V^k$, by the definition of total correlation and Lemma 4.2.1 we have

$$\sum_{T=(w_1,\ldots,w_k)} \left| \boldsymbol{\mu}_S(T) - \prod_{i=1}^{k} x_{u_i,w_i} \right| \leq 2\sqrt{C(\boldsymbol{x}_S)}. \tag{4.3}$$

Therefore,

$$\begin{aligned} & \left| \mathop{\mathbf{E}}_{(S,T)=(u_1,\ldots u_k,v_1,\ldots,v_k)\sim\omega} \left(\boldsymbol{\mu}_S(T) - \prod_{i=1}^{k} x_{u_i,w_i} \right) \right| \\ \leq & \frac{1}{\Delta} \mathop{\mathbf{E}}_{(S,T)=(u_1,\ldots u_k,v_1,\ldots,v_k)\sim V^{2k}} \left| \boldsymbol{\mu}_S(T) - \prod_{i=1}^{k} x_{u_i,w_i} \right| && \text{(by density)} \\ \leq & \frac{1}{\Delta n^k} \mathop{\mathbf{E}}_{S\sim V^k} 2\sqrt{C(\boldsymbol{x}_S)} && \text{(by (4.3))} \end{aligned}$$

$$\leq \frac{2}{\Delta n^k} \sqrt{\mathop{\mathbf{E}}_{S \sim V^k} C(\boldsymbol{x}_S)} \leq n^{-k} \cdot \frac{2\sqrt{\kappa}}{\Delta}. \qquad \text{(by } \kappa\text{-independence)}$$

$\square$

The following variant of Lemma 4.4.2 is used when dealing with the locally dense case later.

Lemma 4.4.3. *Let $\mathcal{I} = (V, \omega)$ be a kAP instance such that $\omega(u_1, \ldots, u_k, w_1, \ldots, w_k) = \omega'(u_1, \ldots, u_k) \cdot \omega''(w_1, \ldots, w_k)$ where ω'' is a Δ-dense distribution over V^k. Let $\boldsymbol{\mu}$ be a κ-independent solution to the k-round Sherali-Adams LP relaxation of $\mathcal{I}$. There is a polynomial-time algorithm, on input $\boldsymbol{\mu}$, to find a solution to $\mathcal{H}$ that certifies that $\mathsf{val}_{\mathcal{H}}(\mathcal{I}) \geq \mathsf{val}_{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}) - 2\sqrt{\kappa}/\Delta$.*

Proof. Let $x_{u,w} = \boldsymbol{\mu}_u(w)$ for all $u, w \in V$. Similar to the proof of Lemma 4.4.2, we have

$$\begin{aligned}
&\left| \mathop{\mathbf{E}}_{(S,T)=(u_1,\ldots u_k, v_1, \ldots, v_k) \sim \omega} \left(\boldsymbol{\mu}_S(T) - \prod_{i=1}^{k} x_{u_i, w_i} \right) \right| \\
\leq &\frac{1}{\Delta} \mathop{\mathbf{E}}_{\substack{S=(u_1,\ldots u_k) \sim \omega' \\ T=(v_1,\ldots,v_k) \sim V^k}} \left| \boldsymbol{\mu}_S(T) - \prod_{i=1}^{k} x_{u_i, w_i} \right| && \text{(by density of } \omega''\text{)} \\
\leq &\frac{1}{\Delta n^k} \mathop{\mathbf{E}}_{S \sim \omega'} 2\sqrt{C(\boldsymbol{x}_S)} && \text{(by (4.3))} \\
\leq &\frac{2}{\Delta n^k} \sqrt{\mathop{\mathbf{E}}_{S \sim \omega'} C(\boldsymbol{x}_S)} \leq n^{-k} \cdot \frac{2\sqrt{\kappa}}{\Delta}. && \text{(by } \kappa\text{-independence)}
\end{aligned}$$

$\square$

4.4.2.2 From relaxation $\mathcal{H}$ to an integral solution

At a first look, $\mathcal{H}$ is very close to the kAP problem itself. However, we cannot independently sample $\pi(v)$ from each $\boldsymbol{x}_v$ in $\mathcal{H}$ to get a solution to kAP, since there is chance that $\pi(v) = \pi(v')$, rendering π not a permutation. Indeed, we show in Section 4.9 that for some kAP instance $\mathcal{I}$, there is a gap between $\mathsf{val}_{\mathcal{H}}(\mathcal{I})$ and $\mathsf{val}(\mathcal{I})$. However, our following lemma shows that this gap cannot be very big for kAP instances $\mathcal{I} = (V, \omega)$ when ω is Δ-dense. The proof is given later in Section 4.6.

Lemma 4.4.4. *Let $\mathcal{I} = (V, \omega)$ be a kAP instance such that ω is Δ-dense. Given a solution $\boldsymbol{x}$ to relaxation $\mathcal{H}$, let $\mathsf{val}_{\mathcal{H}}(\mathcal{I}, \boldsymbol{x})$ be the value of the solution. There is a randomized polynomial-time algorithm to compute a permutation π such that $\mathsf{val}(\mathcal{I}, \pi) \geq \mathsf{val}_{\mathcal{H}}(\mathcal{I}, \boldsymbol{x}) - \frac{7k^2 \log n}{\Delta\sqrt{n}}$.*

In Section 4.7, we prove the following variant of Lemma 4.4.4.

Lemma 4.4.5. *Let $\mathcal{I} = (V, \omega)$ be a kAP instance such that $\omega(u_1, \ldots, u_k, w_1, \ldots, w_k) = \omega'(u_1, \ldots, u_k) \cdot \omega''(w_1, \ldots, w_k)$, where ω' is Δ'-locally dense and ω'' is Δ-dense. Given a solution $\boldsymbol{x}$ to the relaxation $\mathcal{H}$, let $\mathsf{val}_{\mathcal{H}}(\mathcal{I}, \boldsymbol{x})$ be the value of the solution. There is a randomized polynomial-time algorithm to compute a permutation π such that $\mathsf{val}(\mathcal{I}, \pi) \geq \mathsf{val}_{\mathcal{H}}(\mathcal{I}, \boldsymbol{x}) - \frac{7k^2 \log n}{\Delta\Delta'\sqrt{n}}$.*

4.4.3 The rounding lemmas

Combining Lemma 4.4.2 and Lemma 4.4.4, and Lemma 4.4.3 and Lemma 4.4.5, we get the following main rounding lemmas for this subsection.

Lemma 4.4.6. *Let $\mathcal{I} = (V, \omega)$ be a kAP instance such that ω is Δ-dense. Let $\boldsymbol{\mu}$ be a κ-independent solution (with respect to the uniform distribution rather than ω) to the k-round Sherali-Adams LP relaxation for $\mathcal{I}$. There is a polynomial-time algorithm, on input $\boldsymbol{\mu}$, to find a permutation π such that $\mathsf{val}(\mathcal{I}, \pi) \geq \mathsf{val}_{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}) - \frac{2\sqrt{\kappa}}{\Delta} - \frac{7k^2 \log n}{\Delta\sqrt{n}}$.*

Lemma 4.4.7. *Let $\mathcal{I} = (V, \omega)$ be a kAP instance such that $\omega(u_1, \ldots, u_k, w_1, \ldots, w_k) = \omega'(u_1, \ldots, u_k) \cdot \omega''(w_1, \ldots, w_k)$ where ω' is Δ'-locally dense and ω'' is Δ-dense. Let $\boldsymbol{\mu}$ be a κ-independent solution to the k-round Sherali-Adams LP relaxation of $\mathcal{I}$. There is a randomized polynomial-time algorithm, on input $\boldsymbol{\mu}$, to find a permutation π such that $\mathsf{val}(\mathcal{I}, \pi) \geq \mathsf{val}_{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}) - \frac{2\sqrt{\kappa}}{\Delta} - \frac{7k^2 \log n}{\Delta\Delta'\sqrt{n}}$.*

4.5 Putting things together

The following theorem gives PTASs for dense and locally dense MAX-CSP.

Theorem 4.5.1. *Let $\mathcal{I} = (V, \omega, P)$ be a kCSPinstance over finite domain D such that ω is Δ-dense or Δ-locally dense. For any $\epsilon > 0$, let $\ell = \frac{9k^2 4^k \log |D|}{\epsilon^2 \Delta}$. The additive integrality gaps of the $(\ell + k)$-round Sherali-Adams LP relaxation is at most ϵ; and there is a randomized rounding algorithm producing a solution whose value is at least $\mathsf{opt}(\mathcal{I}) - \epsilon$, in expected $n^{O(\ell)}$ time.*

Proof. Let $\boldsymbol{\mu}$ be a solution to the $(\ell + k)$-round Sherali Adams LP relaxation. Let the random variable $(\boldsymbol{\mu}|\boldsymbol{x}_U)$ be the solution after conditioning on the variables in U. By Corollary 4.3.4 and Corollary 4.3.6, we know that there exists $t \le \ell$ such that

$$\mathop{\mathbf{E}}_{U\sim V^t} \sqrt{\mathop{\mathbf{E}}_{S\sim\omega} C_{\boldsymbol{\mu}}(\boldsymbol{x}_S|\boldsymbol{x}_U)} \le \sqrt{\mathop{\mathbf{E}}_{U\sim V^t}\mathop{\mathbf{E}}_{S\sim\omega} C_{\boldsymbol{\mu}}(\boldsymbol{x}_S|\boldsymbol{x}_U)} \le \sqrt{\frac{k^2 4^k \log|D|}{\Delta\ell}} = \frac{\epsilon}{3}.$$

Together with Lemma 4.3.2, we have

$$\mathop{\mathbf{E}}_{U\sim V^t}\left(\mathsf{val}_{\mathrm{LP}}(\mathcal{I},\boldsymbol{\mu}|\boldsymbol{x}_U) - 3\sqrt{\mathop{\mathbf{E}}_{S\sim\omega} C_{\boldsymbol{\mu}}(\boldsymbol{x}_S|\boldsymbol{x}_U)}\right) \ge \mathsf{val}_{\mathrm{LP}}(\mathcal{I},\boldsymbol{\mu}) - \epsilon.$$

We enumerate all the possible ways of conditioning, and find out a solution $\boldsymbol{\mu}'$ to the $(k + \ell - t)$-round Sherali-Adams LP relaxation such that $\mathsf{val}_{\mathrm{LP}}(\mathcal{I},\boldsymbol{\mu}') - 3\sqrt{\mathbf{E}_{S\sim\omega} C_{\boldsymbol{\mu}'}(\boldsymbol{x}_S)} \ge \mathsf{val}_{\mathrm{LP}}(\mathcal{I},\boldsymbol{\mu}) - \epsilon$. Since $\boldsymbol{\mu}'$ is always a $\mathbf{E}_{S\sim\omega} C_{\boldsymbol{\mu}'}(\boldsymbol{x}_S)$-independent solution, by Lemma 4.4.1, given $\boldsymbol{\mu}'$, we can find an assignment with value at least $\mathsf{val}_{\mathrm{LP}}(\mathcal{I},\boldsymbol{\mu}) - \epsilon$ in randomized polynomial time. $\square$

Now we prove that there is a quasi-polynomial-time approximation scheme for dense MAX-AP.

Theorem 4.5.2. *Let $\mathcal{I} = (V,\omega)$ be a kAP instance such that ω is Δ-dense. For any $\epsilon > 0$, let $\ell = \frac{4k4^k\log|D|}{\epsilon^2\Delta^2}$. The additive integrality gaps of the $(\ell + k)$-round Sherali-Adams LP relaxation is at most $\epsilon + \frac{7k^2\log n}{\Delta\sqrt{n}}$; and there is a randomized rounding algorithm producing a solution whose value is at least $\mathsf{opt}(\mathcal{I}) - \epsilon - \frac{7k^2\log n}{\Delta\sqrt{n}}$, in expected $n^{O(\ell)}$ time.*

Proof. Let $\boldsymbol{\mu}$ be a solution to the $(\ell+k)$-round Sherali Adams LP relaxation. By Lemma 4.3.3, we know that there exists $t \le \ell$ such that

$$\mathop{\mathbf{E}}_{U\sim V^t} \sqrt{\mathop{\mathbf{E}}_{S\sim V^k} C_{\boldsymbol{\mu}}(\boldsymbol{x}_S|\boldsymbol{x}_U)} \le \sqrt{\mathop{\mathbf{E}}_{U\sim V^t}\mathop{\mathbf{E}}_{S\sim V^k} C_{\boldsymbol{\mu}}(\boldsymbol{x}_S|\boldsymbol{x}_U)} \le \sqrt{\frac{k4^k\log n}{\ell}} = \frac{\epsilon\Delta}{2}.$$

Together with Lemma 4.3.2, we have

$$\mathop{\mathbf{E}}_{U\sim V^t}\left(\mathsf{val}_{\mathrm{LP}}(\mathcal{I},\boldsymbol{\mu}|\boldsymbol{x}_U) - \frac{2}{\Delta}\sqrt{\mathop{\mathbf{E}}_{S\sim V^k} C_{\boldsymbol{\mu}}(\boldsymbol{x}_S|\boldsymbol{x}_U)}\right) \ge \mathsf{val}_{\mathrm{LP}}(\mathcal{I},\boldsymbol{\mu}) - \epsilon.$$

We enumerate all the possible ways of conditioning, and find out a solution $\boldsymbol{\mu}'$ to the $(k + \ell - t)$-round Sherali-Adams LP relaxation such that $\mathsf{val}_{\mathrm{LP}}(\mathcal{I},\boldsymbol{\mu}') - 2\sqrt{E_{S\sim V^k} C_{\boldsymbol{\mu}'}(\boldsymbol{x}_S)} \ge \mathsf{val}_{\mathrm{LP}}(\mathcal{I},\boldsymbol{\mu}) - \epsilon$. By Lemma 4.4.6, given $\boldsymbol{\mu}'$, we can find a permutation with value at least $\mathsf{val}_{\mathrm{LP}}(\mathcal{I},\boldsymbol{\mu}) - \epsilon - \frac{7k^2\log n}{\Delta\sqrt{n}}$. $\square$

Using Corollary 4.3.6 and Lemma 4.4.7 instead of Lemma 4.3.3 and Lemma 4.4.6, the same argument shows that there is a quasi-polynomial-time approximation scheme for locally dense MAX-AP.

Theorem 4.5.3. *Let $\mathcal{I} = (V, \omega)$ be a kAP instance such that $\omega(u_1, \ldots, u_k, w_1, \ldots, w_k) = \omega'(u_1, \ldots, u_k) \cdot \omega''(w_1, \ldots, w_k)$ where ω' is Δ'-locally dense and ω'' is Δ-dense. For any $\epsilon > 0$, let $\ell = \frac{4k^2 4^k \log|D|}{\epsilon^2 \Delta' \Delta^2}$. The additive integrality gaps of the $(\ell + k)$-round Sherali-Adams LP relaxation is at most $\epsilon + \frac{7k^2 \log n}{\Delta\Delta'\sqrt{n}}$; and there is a randomized rounding algorithm producing a solution whose value is at least $\mathsf{opt}(\mathcal{I}) - \epsilon - \frac{7k^2 \log n}{\Delta\Delta'\sqrt{n}}$, in expected $n^{O(\ell)}$ time.*

4.6 Proof of Lemma 4.4.4

Observe that a solution $\boldsymbol{x}$ to the relaxation $\mathcal{H}$ corresponds to a doubly stochastic matrix. Now let us decompose $\boldsymbol{x}$ into a distribution of permutations $\mathcal{D} = \{\pi : V \to V\}$ such that for any $u, w \in V$, we have $\Pr_{\pi \sim \mathcal{D}}[\pi(u) = w] = \boldsymbol{x}_{u,w}$. Let $\mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{D}) = \mathsf{val}_{\mathcal{H}}(\mathcal{I}, \boldsymbol{x})$ be the value of relaxation $\mathcal{H}$ on $\boldsymbol{x}$ for instance $\mathcal{I}$. Our goal is to "merge" the permutations in $\mathcal{D}$ into one permutation while not losing much in the objective value. The following lemma proves this for the special case when $\mathcal{D}$ is supported on only two permutations.

Lemma 4.6.1. *Let $\mathcal{D}$ be the distribution over π_1 and π_2 such that π_1 is chosen with probability p and π_2 is chosen with probability $(1 - p)$. There exists a distribution $\mathcal{D}'$ over permutations such that for any $k \geq 2$ and any kAP instance $\mathcal{I} = (V, \omega)$ such that ω is Δ-dense , we have*

$$\mathop{\mathbf{E}}_{\pi \sim \mathcal{D}'} [\mathsf{val}(\mathcal{I}, \pi)] \geq \mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{D}) - \frac{2k^2}{\Delta\sqrt{n}}.$$

Moreover, $\mathcal{D}'$ can be sampled in polynomial time.

Proof. Let us assume w.l.o.g. that $V = [n]$, $\pi_1 = \mathrm{id}$ (i.e. $\pi_1(i) = i$ for all $i \in [n]$). For any set $A = \{a_i : a_1 < a_2 < \cdots < a_{|A|} = n\} \subseteq [n]$, let us define π_A be the permutation over $[n]$ so that $\pi_A(i) = a_{t-1} + 1$ if $i = a_t$ for some $t \in [|A|]$ and $\pi_A(i) = i + 1$ otherwise (assuming $a_0 = 0$). We can also assume w.l.o.g. that there exists $A \subseteq [n]$ such that $\pi_2 = \pi_A$. See Figure 4.1. We can add at most $\sqrt{n}$ elements into A to get $A' \subseteq [n]$ such that there is no set of $\sqrt{n}$ consecutive integers that does not intersect A'. It is easy to show that π_A and $\pi_{A'}$ differ at most $2\sqrt{n}$ places. Let $\mathcal{D}_{A'}$ be the probability distribution that chooses π_1 with probability p and $\pi_{A'}$ with probability $(1 - p)$. For any k and any kAP instance $\mathcal{I} = (V, \omega)$ such that ω is Δ-dense, we have

$$\mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{D}) - \mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{D}_{A'})$$

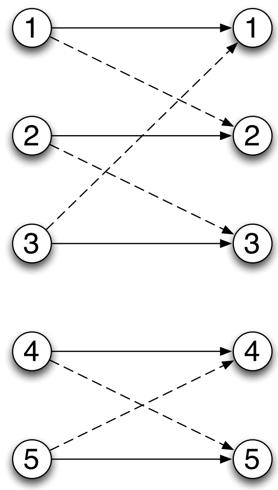

Figure 4.1: Permutations π_1 and π_2 over $[5]$ shown as mappings from $[5]$ to $[5]$.

Solid arrow represents π_1 and dashed arrows represent π_2. For any permutation π_1, we can move vertices in the right side (and relabeling them accordingly) so that the resulting π_1 is the identity permutation. Then for any permutation π_2, we can move pairs of vertices (and relabeling them accordingly) so that π_1 remains the identity permutation whereas the cycles formed by π_1 and π_2 are drawn disjointly. In such a case, π_2 satisfies the condition in the body text.

$$
\begin{aligned}
&= n^k \mathop{\mathbf{E}}_{(U,W)\sim\omega} \left(\prod_{i=1}^{k} \Pr_{\pi\sim\mathcal{D}}[\pi(u_i) = w_i] - \prod_{i=1}^{k} \Pr_{\pi\sim\mathcal{D}_{A'}}[\pi(u_i) = w_i] \right) \\
&\leq n^k \mathop{\mathbf{E}}_{(U,W)\sim\omega} \mathbf{1}\left[\exists i \in [k] : \pi_{A'}(u_i) \neq \pi_A(u_i)\right] \cdot \prod_{i=1}^{k} \Pr_{\pi\sim\mathcal{D}}[\pi(u_i) = w_i] \\
&\leq \frac{n^k}{\Delta} \cdot \mathop{\mathbf{E}}_{(U,W)\sim V^{2k}} \mathbf{1}\left[\exists i \in [k] : \pi_{A'}(u_i) \neq \pi_A(u_i)\right] \cdot \prod_{i=1}^{k} \Pr_{\pi\sim\mathcal{D}}[\pi(u_i) = w_i], \qquad (4.4)
\end{aligned}
$$

where the last inequality is by the density of ω.

Since

$$
n^k \mathop{\mathbf{E}}_{W\sim V^k} \prod_{i=1}^{k} \Pr_{\pi\sim\mathcal{D}}[\pi(u_i) = w_i] = \prod_{i=1}^{k} \sum_{w_i\sim V} \Pr_{\pi\sim\mathcal{D}}[\pi(u_i) = w_i] = 1, \qquad (4.5)
$$

we have

$$(4.4) \leq \frac{1}{\Delta} \cdot \Pr_{U \sim V^k} [\exists i \in [k] : \pi_{A'}(u_i) \neq \pi_A(u_i)] \leq \frac{2k}{\Delta\sqrt{n}}. \tag{4.6}$$

Now we define the distribution $\mathcal{D}'$. Let us assume that the elements in A' are $a'_1 < a'_2 < \ldots < a'_{|A'|} = n$; let $a'_0 = 0$ for convenience. To draw a permutation $\pi \sim \mathcal{D}'$, we sample $|A'|$ i.i.d. $0/1$ bits $b_1, b_2, \ldots, b_{|A'|}$, each of which has mean p. For each i, we find out the unique $t \in [|A'|]$ so that $a'_{t-1} < i \leq a'_t$; let $\pi(i) = \pi_1(i) = i$ if $b_t = 0$; let $\pi(i) = \pi_{A'}(i)$ otherwise.

For any k and any Δ-dense kAP instance $\mathcal{I} = (V, \omega)$, we have

$$\begin{aligned}
&\mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{D}_{A'}) - \mathop{\mathbf{E}}_{\pi \sim \mathcal{D}'} \mathsf{val}(\mathcal{I}, \pi) \\
=& n^k \mathop{\mathbf{E}}_{(U,W) \sim \omega} \left(\prod_{i=1}^{k} \Pr_{\pi \sim \mathcal{D}_{A'}} [\pi(u_i) = w_i] - \Pr_{\pi \sim \mathcal{D}'} [\forall i \in [k] : \pi(u_i) = w_i] \right) \\
=& n^k \mathop{\mathbf{E}}_{(U,W) \sim \omega} \left(\prod_{i=1}^{k} \Pr_{\pi \sim \mathcal{D}_{A'}} [\pi(u_i) = w_i] \right. \\
&\qquad \left. - \prod_{t=1}^{|A'|} \Pr_{\pi \sim \mathcal{D}'} [\forall i \in [k], a'_{t-1} < u_i \leq a'_t : \pi(u_i) = w_i] \right) \\
\leq& n^k \mathop{\mathbf{E}}_{(U,W) \sim \omega} \mathbf{1} \left[\exists t \in [|A'|] : \exists \text{ more than one } i \text{ s.t. } a'_{t-1} < u_i \leq a'_t \right] \\
&\qquad \cdot \prod_{i=1}^{k} \Pr_{\pi \sim \mathcal{D}_{A'}} [\pi(u_i) = w_i] \\
\leq& \frac{n^k}{\Delta} \mathop{\mathbf{E}}_{(U,W) \sim V^{2k}} \mathbf{1} \left[\exists t \in [|A'|] : \exists \text{ more than one } i \text{ s.t. } a'_{t-1} < u_i \leq a'_t \right] \\
&\qquad \cdot \prod_{i=1}^{k} \Pr_{\pi \sim \mathcal{D}_{A'}} [\pi(u_i) = w_i] \qquad \text{(by density)} \\
=& \frac{1}{\Delta} \Pr_{U \sim V^k} \left[\exists t \in [|A'|] : \exists \text{ more than one } i \text{ s.t. } a'_{t-1} < u_i \leq a'_t \right] \qquad \text{(by (4.5))} \\
\leq& \frac{1}{\Delta} \cdot \binom{k}{2} \cdot \sum_{t \in [|A'|]} \left(\frac{a'_t - a'_{t-1}}{n} \right)^2 \leq \frac{1}{\Delta} \cdot \binom{k}{2} \cdot \sum_{t \in [|A'|]} \left(\frac{a'_t - a'_{t-1}}{n} \right) \cdot \frac{1}{\sqrt{n}} \leq \frac{k^2}{\sqrt{n}}. \qquad (4.7)
\end{aligned}$$

The lemma is proved by combining (4.6) and (4.7). $\square$

of Lemma 4.4.4. Let $\mathcal{D}$ be supported on $\pi_1, \pi_2, \dots, \pi_m$, each π_i is chosen with probability p_i. We can assume that $m \le n^2$ by preserving only the n^2 permutations with the largest probabilities and proper normalization, which would cause a loss of at most n^{-1} in the objective value $\mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{D})$. Now we show that for any such distribution, we can find a distribution $\mathcal{E}$ that is supported on $(m-1)$ permutations, such that

$$\mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{E}) \ge \mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{D}) - (p_1 + p_2)\frac{2k^2}{\Delta\sqrt{n}}. \tag{4.8}$$

In other words, since π_1 and π_2 are arbitrary, we are able to "merge" any two permutations π_i and π_j in $\mathcal{D}$ by paying a loss of $(p_i+p_j)\frac{2k^2}{\Delta\sqrt{n}}$ in the objective value. We repeatedly merge the two permutations with the smallest probability mass in the distribution until there is only one permutation left, during this process we lose at most $\lceil \log m \rceil \frac{2k^2}{\Delta\sqrt{n}} \le \frac{6k^2 \log n}{\Delta\sqrt{n}}$ in objective value. Together with the n^{-1} loss at the beginning of the proof, we lose at most $\frac{7k^2 \log n}{\Delta\sqrt{n}}$ for sufficiently large n.

In order to show (4.8), let us define a distribution $\tilde{\mathcal{E}}$ of distributions of permutations as follows. Let $\mathcal{F}$ to be the distribution of permutations that chooses π_1 with probability $\frac{p_1}{p_1+p_2}$ and π_2 with the remaining probability. Apply Lemma 4.6.1 on $\mathcal{F}$ to get $\mathcal{F}'$. A distribution $\mathcal{E}$ from $\tilde{\mathcal{E}}$ is sampled by first sampling a permutation π from $\mathcal{F}'$, and returning the distribution that puts probability mass (p_1+p_2) on π and p_i on π_i for all $i : 3 \le i \le m$. For every $u, w \in V$, let $\gamma_{u,w} = \sum_{t=3}^{m} p_t \mathbf{1}[\pi_t(u) = w] \le 1$. We have

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\mathcal{E}\sim\tilde{\mathcal{E}}} \mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{E}) = \mathop{\mathbf{E}}_{\mathcal{E}\sim\tilde{\mathcal{E}}} \mathop{\mathbf{E}}_{(U,W)\sim\omega} \prod_{i=1}^{k} \Pr_{\pi\sim\mathcal{E}}[\pi(u_i) = w_i] \\
&= \mathop{\mathbf{E}}_{\pi\sim\mathcal{F}'} \mathop{\mathbf{E}}_{(U,W)\sim\omega} \prod_{i=1}^{k} \left((p_1+p_2)\mathbf{1}[\pi(u_i) = w_i] + \gamma_{u_i,w_i}\right) \\
&= \sum_{Q\subseteq[k]} \mathop{\mathbf{E}}_{(U,W)\sim\omega} \left(\mathop{\mathbf{E}}_{\pi\sim\mathcal{F}'} \prod_{i\in Q} (p_1+p_2)\mathbf{1}[\pi(u_i) = w_i] \right) \left(\prod_{i\in\overline{Q}} \gamma_{u_i,w_i} \right) \\
&= \sum_{Q\subseteq[k]} \mathop{\mathbf{E}}_{\substack{(U_{\overline{Q}},W_{\overline{Q}}) \\ \sim\omega_{(\overline{Q},\overline{Q})}}} \left(\prod_{i\in\overline{Q}} \gamma_{u_i,w_i} \right) (p_1+p_2)^{|Q|} \mathop{\mathbf{E}}_{\substack{(U_Q,W_Q) \\ \sim\omega|(U_{\overline{Q}},W_{\overline{Q}})}} \left(\mathop{\mathbf{E}}_{\pi\sim\mathcal{F}'} \prod_{i\in Q} \mathbf{1}[\pi(u_i) = w_i] \right),
\end{aligned} \tag{4.9}$$

where U_A is the restriction of vector U over coordinates in A, $\omega_{(A,B)}$ is the marginal distribution of ω over A in the first k coordinates and B in the last k coordinates, and $\omega|(U_A, W_B)$ is the distribution ω conditioned on that coordinates in A are assigned U and coordinates in B are assigned W. Let $\mathcal{I}_{U_{\overline{Q}},W_{\overline{Q}}}$ be the $|Q|$-AP instance $\big(V,\omega|(U_{\overline{Q}},W_{\overline{Q}})\big)$. We know that $\omega|(U_{\overline{Q}},W_{\overline{Q}})$ is $\Delta\cdot n^{2|\overline{Q}|}\cdot\omega_{(\overline{Q},\overline{Q})}(U_{\overline{Q}},W_{\overline{Q}})$-dense. Therefore for every $Q\neq\emptyset$, by Lemma 4.6.1, we have

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{\substack{(U_Q,W_Q)\\ \sim\omega|(U_{\overline{Q}},W_{\overline{Q}})}}\left(\mathop{\mathbf{E}}_{\pi\sim\mathcal{F}'}\prod_{i\in Q}\mathbf{1}[\pi(u_i)=w_i]\right)=\mathop{\mathbf{E}}_{\pi\sim\mathcal{F}'}\mathsf{val}(\mathcal{I}_{U_{\overline{Q}},W_{\overline{Q}}},\pi)\\
\geq&\mathsf{val}_{\mathcal{H}}(\mathcal{I}_{U_{\overline{Q}},W_{\overline{Q}}},\mathcal{F})-\frac{2k^2n^{-2|\overline{Q}|}}{\Delta\omega_{(\overline{Q},\overline{Q})}(U_{\overline{Q}},W_{\overline{Q}})\sqrt{n}}\\
=&\mathop{\mathbf{E}}_{\substack{(U_Q,W_Q)\\ \sim\omega|(U_{\overline{Q}},W_{\overline{Q}})}}\prod_{i\in Q}\left(\frac{p_1\mathbf{1}[\pi_1(u_i)=w_i]+p_2\mathbf{1}[\pi_2(u_i)=w_i]}{p_1+p_2}\right)-\frac{2k^2n^{-2|\overline{Q}|}}{\Delta\omega_{(\overline{Q},\overline{Q})}(U_{\overline{Q}},W_{\overline{Q}})\sqrt{n}}.
\end{aligned}
$$

Therefore we have

$$
\begin{aligned}
(4.9)\geq&\sum_{Q\subseteq[k]}\mathop{\mathbf{E}}_{\substack{(U_{\overline{Q}},W_{\overline{Q}})\\ \sim\omega_{(\overline{Q},\overline{Q})}}}\left(\prod_{i\in\overline{Q}}\gamma_{u_i,w_i}\right)\mathop{\mathbf{E}}_{\substack{(U_Q,W_Q)\\ \sim\omega|(U_{\overline{Q}},W_{\overline{Q}})}}\prod_{i\in Q}(p_1\mathbf{1}[\pi_1(u_i)=w_i]+p_2\mathbf{1}[\pi_2(u_i)=w_i])\\
&-\sum_{\emptyset\neq Q\subseteq[k]}\mathop{\mathbf{E}}_{\substack{(U_{\overline{Q}},W_{\overline{Q}})\\ \sim\omega_{(\overline{Q},\overline{Q})}}}\left(\prod_{i\in\overline{Q}}\gamma_{u_i,w_i}\right)(p_1+p_2)^{|Q|}\cdot\frac{2k^2n^{-2|\overline{Q}|}}{\Delta\omega_{(\overline{Q},\overline{Q})}(U_{\overline{Q}},W_{\overline{Q}})\sqrt{n}}\\
=&\mathop{\mathbf{E}}_{(U,W)\sim\omega}\left(\sum_{t=1}^{m}p_t\mathbf{1}[\pi_t(u_i)=w_i]\right)\\
&-\sum_{\emptyset\neq Q\subseteq[k]}\mathop{\mathbf{E}}_{\substack{(U_{\overline{Q}},W_{\overline{Q}})\\ \sim V^{\overline{Q}\times\overline{Q}}}}\left(\prod_{i\in\overline{Q}}\gamma_{u_i,w_i}\right)(p_1+p_2)^{|Q|}\cdot\frac{2k^2}{\Delta\sqrt{n}}\\
\geq&\mathsf{val}_{\mathcal{H}}(\mathcal{I},\mathcal{D})-(p_1+p_2)\frac{2k^2}{\Delta\sqrt{n}}.
\end{aligned}
$$

In all, we have proved that $\mathbf{E}_{\mathcal{E}\sim\tilde{\mathcal{E}}}\,\mathsf{val}_{\mathcal{H}}(\mathcal{I},\mathcal{E})\geq\mathsf{val}_{\mathcal{H}}(\mathcal{I},\mathcal{D})-(p_1+p_2)\frac{2k^2}{\Delta\sqrt{n}}$. Since $\tilde{\mathcal{E}}$ can be sampled in polynomial time, there is a randomized polynomial-time algorithm to find out a $\mathcal{E}$ satisfying (4.8). $\square$

4.7 Proof of Lemma 4.4.5

We say a distribution ω' over V^k is Δ'-*well spread* if for every $i, j \in [k]$ such that $i \neq j$, and for every disjoint partition $V = V_1 \cup V_2 \cup \cdots \cup V_t$, we have

$$\Delta' \cdot \Pr_{(u_1,\ldots,u_k)\sim\omega'}[\exists t' \in [t] : u_i \in V_{t'} \text{ and } u_j \in V_{t'}] \leq \frac{\max_{t'\in[t]} |V_{t'}|}{n}.$$

Claim 4.7.1. *A Δ'-locally dense distribution ω' is (Δ'/k)-well spread.*

Proof. W.l.o.g. we assume that $i = 1$ and $j = 2$. For every $Z \subseteq V$, we have

$$\begin{aligned}
&\Pr_{(u_1,\ldots,u_k)\sim\omega'}[\exists t' \in [t] : u_i \in V_{t'} \text{ and } u_j \in V_{t'}] \\
&= \sum_{t'\in[t]} \sum_{\substack{u_1,u_2\in V_{t'}\\ u_3,\ldots,u_k\in V}} \omega'(u_1,\ldots,u_k) \leq \sum_{t'\in[t]} \sum_{\substack{u_1,u_2\in V_{t'}\\ u_3,\ldots,u_k\in V}} \frac{\sum_{i=1}^k d_i(u_i)}{\Delta' n^{k-1}} \\
&= \sum_{t'\in[t]} \left(\sum_{\substack{u_2\in V_{t'}\\ u_3,\ldots,u_k\in V}} \frac{\sum_{u_1\in V_{t'}} d_1(u_1)}{\Delta' n^{k-1}} + \sum_{\substack{u_1\in V_{t'}\\ u_3,\ldots,u_k\in V}} \frac{\sum_{u_2\in V_{t'}} d_2(u_2)}{\Delta' n^{k-1}} \right. \\
&\qquad\qquad \left. + \sum_{i=3}^{k} \sum_{\substack{u_1,u_2\in V_{t'}\\ u_3,\ldots,u_{i-1},u_{i+1},\ldots,u_k\in V}} \frac{\sum_{u_i\in V} d_i(u_i)}{\Delta' n^{k-1}} \right) \\
&\leq 2 \cdot \frac{\max_{t'\in[t]} |V_{t'}|}{\Delta' n} + (k-2) \cdot \frac{\Delta' \max_{t'\in[t]} |V_{t'}|^2}{n^2} \leq \frac{k \max_{t'\in[t]} |V_{t'}|}{\Delta' n}.
\end{aligned}$$

□

We will prove a slightly stronger statement than that of Lemma 4.4.5, in the sense that we prove the lemma for every ω such that $\omega = \omega' \cdot \omega''$ where ω' is Δ'-well spread and ω'' is Δ-dense.

The proof goes along the lines of the proof of Lemma 4.4.4. We decompose x into a distribution of permutations $\mathcal{D} = \{\pi : V \to V\}$ such that for any $u, w \in V$, we have $\Pr_{\pi\sim\mathcal{D}}[\pi(u) = w] = x_{u,w}$. We first prove following lemma, which is an analogy of Lemma 4.6.1.

Lemma 4.7.2. *Let $\mathcal{D}$ be the distribution over π_1 and π_2 such that π_1 is chosen with probability p and π_2 is chosen with probability $(1-p)$. There exists a distribution $\mathcal{D}'$ over* *permutations and a distribution $\mathcal{V}$ over the disjoint partitions $\{(V_1 \cup \cdots \cup V_t)\}$ where each V_i has at most $2\sqrt{n}$ elements, such that for any $k \geq 2$ and any kAP instance $\mathcal{I} = (V, \omega)$ such that $\omega = \omega' \cdot \omega''$ where ω'' is Δ-dense, we have*

$$\mathop{\mathbf{E}}_{\pi\sim\mathcal{D}'}[\mathsf{val}(\mathcal{I},\pi)] \geq \mathsf{val}_{\mathcal{H}}(\mathcal{I},\mathcal{D}) - \frac{2k}{\Delta\sqrt{n}} - \frac{1}{\Delta}\sum_{1\leq i<j\leq k}\mathop{\mathbf{E}}_{(V_1\cup\ldots\cup V_t)\sim\mathcal{V}}\sum_{t'\in[t]}\Pr_{(u_1,\ldots,u_k)\sim\omega'}[u_i \in V_{t'} \text{ and } u_j \in V_{t'}].$$

Moreover, $\mathcal{D}'$ can be sampled in polynomial time.

Proof. Let us assume w.l.o.g. that $V = [n]$, $\pi_1 = \mathrm{id}$ (i.e. $\pi_1(i) = i$ for all $i \in [n]$). For any set $A = \{a_i : a_1 < a_2 < \cdots < a_{|A|} = n\} \subseteq [n]$, let us define π_A be the permutation over $[n]$ so that $\pi_A(i) = a_{t-1} + 1$ if $i = a_t$ for some $t \in [|A|]$ and $\pi_A(i) = i + 1$ otherwise (assuming $a_0 = 0$). We can also assume w.l.o.g. that there exists $A \subseteq [n]$ such that $\pi_2 = \pi_A$.

Now we define the random set variable $A' : A \subseteq A \subseteq V$ as follows. We start from $A' = A$, and for each $i \in [\sqrt{n}]$, we uniformly sample an element a from $((i-1)\sqrt{n}, i\sqrt{n}]$ and let $A' \leftarrow A' \cup \{a\}$. In this way, we know that there is no set of $2\sqrt{n}$ consecutive integers that does not intersect A'. It is easy to show that for every $v \in V$, $\Pr_{A'}[\pi_A(v) \neq \pi_{A'}(v)] \leq \frac{2}{\sqrt{n}}$.

Let $\mathcal{D}_{A'}$ be the probability distribution that chooses π_1 with probability p and $\pi_{A'}$ with probability $(1-p)$. For any k and any kAP instance $\mathcal{I} = (V, \omega)$ such that ω is Δ-dense, we have

$$\begin{aligned}
&\mathsf{val}_{\mathcal{H}}(\mathcal{I},\mathcal{D}) - \mathop{\mathbf{E}}_{A'}\mathsf{val}_{\mathcal{H}}(\mathcal{I},\mathcal{D}_{A'}) \\
=& n^k \mathop{\mathbf{E}}_{(U,W)\sim\omega}\left(\prod_{i=1}^{k}\Pr_{\pi\sim\mathcal{D}}[\pi(u_i) = w_i] - \prod_{i=1}^{k}\Pr_{\pi\sim\mathcal{D}_{A'}}[\pi(u_i) = w_i]\right) \\
\leq& n^k \mathop{\mathbf{E}}_{U\sim\omega',W\sim\omega''}\mathop{\mathbf{E}}_{A'}\mathbf{1}\left[\exists i \in [k] : \pi_{A'}(u_i) \neq \pi_A(u_i)\right] \cdot \prod_{i=1}^{k}\Pr_{\pi\sim\mathcal{D}}[\pi(u_i) = w_i] \\
\leq& \frac{n^k}{\Delta} \cdot \mathop{\mathbf{E}}_{U\sim\omega'}\mathop{\mathbf{E}}_{A'}\mathbf{1}\left[\exists i \in [k] : \pi_{A'}(u_i) \neq \pi_A(u_i)\right] \cdot \mathop{\mathbf{E}}_{W\sim V^k}\prod_{i=1}^{k}\Pr_{\pi\sim\mathcal{D}}[\pi(u_i) = w_i],
\end{aligned} \tag{4.10}$$

where the last inequality is by the density of ω''. By (4.5), we have

$$(4.10) \leq \frac{1}{\Delta} \cdot \mathop{\mathbf{E}}_{U\sim\omega'}\Pr_{A'}\left[\exists i \in [k] : \pi_{A'}(u_i) \neq \pi_A(u_i)\right] \leq \frac{2k}{\Delta\sqrt{n}}. \tag{4.11}$$

For every $A' \subseteq [n]$, we define the distribution $\mathcal{D}'_{A'}$. Let us assume that the elements in A' are $a'_1 < a'_2 < \ldots < a'_{|A'|} = n$; let $a'_0 = 0$ for convenience. To draw a permutation $\pi \sim \mathcal{D}'_{A'}$, we sample $|A'|$ i.i.d. $0/1$ bits $b_1, b_2, \ldots, b_{|A'|}$, each of which has mean p. For each i, we find out the unique $t \in [|A'|]$ so that $a'_{t-1} < i \leq a'_t$; let $\pi(i) = \pi_1(i) = i$ if $b_t = 0$; let $\pi(i) = \pi_{A'}(i)$ otherwise.

Now we define the distribution $\mathcal{D}'$. To draw a permutation $\pi \sim \mathcal{D}'$, we first sample a random set A', and then draw a permutation from $\mathcal{D}'_{A'}$.

For any k and any kAP instance $\mathcal{I} = (V, \omega)$ such that $\omega = \omega' \cdot \omega''$ where ω'' is Δ-dense, we have

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{A'} \mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{D}_{A'}) - \mathop{\mathbf{E}}_{\pi \sim \mathcal{D}'} \mathsf{val}(\mathcal{I}, \pi) \\
=& n^k \mathop{\mathbf{E}}_{(U,W)\sim\omega} \mathop{\mathbf{E}}_{A'} \left(\prod_{i=1}^{k} \Pr_{\pi \sim \mathcal{D}_{A'}} [\pi(u_i) = w_i] - \Pr_{\pi \sim \mathcal{D}'_{A'}} [\forall i \in [k] : \pi(u_i) = w_i] \right) \\
=& n^k \mathop{\mathbf{E}}_{(U,W)\sim\omega} \mathop{\mathbf{E}}_{A'} \left(\prod_{i=1}^{k} \Pr_{\pi \sim \mathcal{D}_{A'}} [\pi(u_i) = w_i] \right. \\
& \qquad\qquad\qquad\qquad \left. - \prod_{t=1}^{|A'|} \Pr_{\pi \sim \mathcal{D}'_{A'}} [\forall i \in [k], a'_{t-1} < u_i \leq a'_t : \pi(u_i) = w_i] \right) \\
\leq& n^k \mathop{\mathbf{E}}_{(U,W)\sim\omega} \mathop{\mathbf{E}}_{A'} \mathbf{1} \left[\exists t \in [|A'|] : \exists \text{ more than one } i \text{ s.t. } a'_{t-1} < u_i \leq a'_t \right] \\
& \qquad\qquad\qquad\qquad \cdot \prod_{i=1}^{k} \Pr_{\pi \sim \mathcal{D}_{A'}} [\pi(u_i) = w_i] \\
\leq& \frac{n^k}{\Delta} \mathop{\mathbf{E}}_{U \sim \omega'} \mathop{\mathbf{E}}_{A'} \mathbf{1} \left[\exists t \in [|A'|] : \exists \text{ more than one } i \text{ s.t. } a'_{t-1} < u_i \leq a'_t \right] \\
& \qquad\qquad\qquad\qquad \cdot \mathop{\mathbf{E}}_{W \sim V^k} \prod_{i=1}^{k} \Pr_{\pi \sim \mathcal{D}_{A'}} [\pi(u_i) = w_i] \qquad \text{(by density)} \\
\leq& \frac{1}{\Delta} \mathop{\mathbf{E}}_{U \sim \omega'} \mathop{\mathbf{E}}_{A'} \mathbf{1} \left[\exists t \in [|A'|] : \exists \text{ more than one } i \text{ s.t. } a'_{t-1} < u_i \leq a'_t \right] \qquad \text{(by (4.5))} \\
\leq& \frac{1}{\Delta} \sum_{1 \leq i < j \leq k} \mathop{\mathbf{E}}_{A'} \sum_{t \in [|A'|]} \Pr_{(u_1, \ldots, u_k) \sim \omega'} [a'_{t-1} < u_i, u_j \leq a'_t]. \qquad (4.12)
\end{aligned}
$$

The lemma is proved by combining (4.11) and (4.12). $\square$

Now we are ready to prove Lemma 4.4.5.

of Lemma 4.4.5. Let $\mathcal{D}$ be supported on $\pi_1, \pi_2, \ldots, \pi_m$, each π_i is chosen with probability p_i. We can assume that $m \le n^2$ by preserving only the n^2 permutations with the largest probabilities and proper normalization, which would cause a loss of at most n^{-1} in the objective value $\mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{D})$. Now we show that for any such distribution, we can find a distribution $\mathcal{E}$ that is supported on $(m-1)$ permutations, such that

$$\mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{E}) \ge \mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{D}) - (p_1 + p_2)\frac{2k^2}{\Delta\Delta'\sqrt{n}}. \tag{4.13}$$

In other words, since π_1 and π_2 are arbitrary, we are able to "merge" any two permutations π_i and π_j in $\mathcal{D}$ by paying a loss of $(p_i + p_j)\frac{2k^2}{\Delta\Delta'\sqrt{n}}$ in the objective value. We repeatedly merge the two permutations with the smallest probability mass in the distribution until there is only one permutation left, during this process we lose at most $\lceil \log m \rceil \frac{2k^2}{\Delta\sqrt{n}} \le \frac{6k^2 \log n}{\Delta\Delta'\sqrt{n}}$ in objective value. Together with the n^{-1} loss at the beginning of the proof, we lose at most $\frac{7k^2 \log n}{\Delta\Delta'\sqrt{n}}$ for sufficiently large n.

In order to show (4.13), let us define a distribution $\tilde{\mathcal{E}}$ of distributions of permutations as follows. Let $\mathcal{F}$ to be the distribution of permutations that chooses π_1 with probability $\frac{p_1}{p_1+p_2}$ and π_2 with the remaining probability. Apply Lemma 4.7.2 on $\mathcal{F}$ to get $\mathcal{F}'$. A distribution $\mathcal{E}$ from $\tilde{\mathcal{E}}$ is sampled by first sampling a permutation π from $\mathcal{F}'$, and returning the distribution that puts probability mass (p_1+p_2) on π and p_i on π_i for all $i : 3 \le i \le m$. For every $u, w \in V$, let $\gamma_{u,w} = \sum_{t=3}^{m} p_t \mathbf{1}[\pi_t(u) = w] \le 1$. We have

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\mathcal{E}\sim\tilde{\mathcal{E}}} \mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{E}) = \mathop{\mathbf{E}}_{\mathcal{E}\sim\tilde{\mathcal{E}}} \mathop{\mathbf{E}}_{(U,W)\sim\omega} \prod_{i=1}^{k} \Pr_{\pi\sim\mathcal{E}}[\pi(u_i) = w_i] \\
&= \mathop{\mathbf{E}}_{\pi\sim\mathcal{F}'} \mathop{\mathbf{E}}_{(U,W)\sim\omega} \prod_{i=1}^{k} \left((p_1 + p_2)\mathbf{1}[\pi(u_i) = w_i] + \gamma_{u_i,w_i}\right) \\
&= \sum_{Q\subseteq[k]} \mathop{\mathbf{E}}_{(U,W)\sim\omega} \left(\mathop{\mathbf{E}}_{\pi\sim\mathcal{F}'} \prod_{i\in Q}(p_1 + p_2)\mathbf{1}[\pi(u_i) = w_i]\right)\left(\prod_{i\in\overline{Q}} \gamma_{u_i,w_i}\right) \\
&= \sum_{Q\subseteq[k]} \mathop{\mathbf{E}}_{\substack{(U_{\overline{Q}},W_{\overline{Q}})\\ \sim\omega_{(\overline{Q},\overline{Q})}}} \left(\prod_{i\in\overline{Q}} \gamma_{u_i,w_i}\right)(p_1 + p_2)^{|Q|} \mathop{\mathbf{E}}_{\substack{(U_Q,W_Q)\\ \sim\omega|(U_{\overline{Q}},W_{\overline{Q}})}} \left(\mathop{\mathbf{E}}_{\pi\sim\mathcal{F}'} \prod_{i\in Q} \mathbf{1}[\pi(u_i) = w_i]\right),
\end{aligned} \tag{4.14}$$

where U_A is the restriction of vector U over coordinates in A, ω''_A is the marginal distribution of ω'' over the coordinates in A, and $\omega_{(A,B)}$ is the marginal distribution of ω over A in the first k coordinates and B in the last k coordinates. Let $\mathcal{I}_{U_{\overline{Q}},W_{\overline{Q}}}$ be the $|Q|$-AP

instance $\left(V, \omega|(U_{\overline{Q}}, W_{\overline{Q}})\right)$. We know that $\omega|(U_{\overline{Q}}, W_{\overline{Q}}) = (\omega'|U_{\overline{Q}}) \cdot (\omega''|W_{\overline{Q}})$, and $\omega''|W_{\overline{Q}}$ is $\Delta \cdot n^{|\overline{Q}|} \cdot \omega''_{\overline{Q}}(W_{\overline{Q}})$-dense. Therefore for every $Q \neq \emptyset$, by Lemma 4.7.2, we have

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\substack{(U_Q,W_Q)\\ \sim\omega|(U_{\overline{Q}},W_{\overline{Q}})}} \left(\mathop{\mathbf{E}}_{\pi\sim\mathcal{F}'} \prod_{i\in Q} \mathbf{1}[\pi(u_i) = w_i] \right) = \mathop{\mathbf{E}}_{\pi\sim\mathcal{F}'} \mathsf{val}(\mathcal{I}_{U_{\overline{Q}},W_{\overline{Q}}}, \pi) \\
&\geq \mathsf{val}_{\mathcal{H}}(\mathcal{I}_{U_{\overline{Q}},W_{\overline{Q}}}, \mathcal{F}) - \frac{2kn^{-|\overline{Q}|}}{\Delta\omega''_{\overline{Q}}(W_{\overline{Q}})\sqrt{n}} \\
&\qquad - \frac{n^{-|\overline{Q}|}}{\Delta\omega''_{\overline{Q}}(W_{\overline{Q}})} \sum_{\substack{i,j\in Q\\ i\neq j}} \mathop{\mathbf{E}}_{(V_1\cup\ldots\cup V_t)\sim\mathcal{V}} \sum_{t'\in[t]} \Pr_{U_Q\sim\omega'|U_{\overline{Q}}}[u_i \in V_{t'} \text{ and } u_j \in V_{t'}] \\
&= \mathop{\mathbf{E}}_{\substack{(U_Q,W_Q)\\ \sim\omega|(U_{\overline{Q}},W_{\overline{Q}})}} \prod_{i\in Q} \left(\frac{p_1\mathbf{1}[\pi_1(u_i) = w_i] + p_2\mathbf{1}[\pi_2(u_i) = w_i]}{p_1 + p_2} \right) - \frac{2kn^{-|\overline{Q}|}}{\Delta\omega''_{\overline{Q}}(W_{\overline{Q}})\sqrt{n}} \\
&\qquad - \frac{n^{-|\overline{Q}|}}{\Delta\omega''_{\overline{Q}}(W_{\overline{Q}})} \sum_{\substack{i,j\in Q\\ i\neq j}} \mathop{\mathbf{E}}_{(V_1\cup\ldots\cup V_t)\sim\mathcal{V}} \sum_{t'\in[t]} \Pr_{U_Q\sim\omega'|U_{\overline{Q}}}[u_i \in V_{t'} \text{ and } u_j \in V_{t'}].
\end{aligned}$$

Therefore we have

(4.14)

$$\begin{aligned}
&\geq \sum_{Q\subseteq[k]} \mathop{\mathbf{E}}_{\substack{(U_{\overline{Q}},W_{\overline{Q}})\\ \sim\omega_{(\overline{Q},\overline{Q})}}} \left(\prod_{i\in\overline{Q}} \gamma_{u_i,w_i} \right) \mathop{\mathbf{E}}_{\substack{(U_Q,W_Q)\\ \sim\omega|(U_{\overline{Q}},W_{\overline{Q}})}} \prod_{i\in Q} \left(p_1\mathbf{1}[\pi_1(u_i) = w_i] + p_2\mathbf{1}[\pi_2(u_i) = w_i]\right) \\
&\quad - \sum_{\emptyset\neq Q\subseteq[k]} \mathop{\mathbf{E}}_{\substack{(U_{\overline{Q}},W_{\overline{Q}})\\ \sim\omega_{(\overline{Q},\overline{Q})}}} \left(\prod_{i\in\overline{Q}} \gamma_{u_i,w_i} \right) (p_1 + p_2)^{|Q|} \cdot \frac{2kn^{-|\overline{Q}|}}{\Delta\omega''_{\overline{Q}}(W_{\overline{Q}})\sqrt{n}} \\
&\quad - \sum_{\emptyset\neq Q\subseteq[k]} \mathop{\mathbf{E}}_{\substack{(U_{\overline{Q}},W_{\overline{Q}})\\ \sim\omega_{(\overline{Q},\overline{Q})}}} \left(\prod_{i\in\overline{Q}} \gamma_{u_i,w_i} \right) (p_1 + p_2)^{|Q|} \\
&\qquad\qquad\qquad \cdot \frac{n^{|\overline{Q}|}}{\Delta\omega''_{\overline{Q}}(W_{\overline{Q}})} \sum_{\substack{i,j\in Q\\ i\neq j}} \mathop{\mathbf{E}}_{\substack{(V_1\cup\ldots\cup V_t)\\ \sim\mathcal{V}}} \sum_{t'\in[t]} \Pr_{\substack{U_Q\sim\\ \omega'|U_{\overline{Q}}}} [u_i, u_j \in V_{t'}] \\
&= \mathop{\mathbf{E}}_{(U,W)\sim\omega} \left(\sum_{t=1}^{m} p_t \mathbf{1}[\pi_t(u_i) = w_i] \right)
\end{aligned}$$

$$- \sum_{\emptyset \neq Q \subseteq [k]} \mathop{\mathbf{E}}_{\substack{U_{\overline{Q}} \sim \omega'_{\overline{Q}} \\ W_{\overline{Q}} \sim V^{\overline{Q}}}} \left(\prod_{i \in \overline{Q}} \gamma_{u_i, w_i} \right) (p_1 + p_2)^{|Q|} \cdot \frac{2k}{\Delta \sqrt{n}}$$

$$- \sum_{\emptyset \neq Q \subseteq [k]} \mathop{\mathbf{E}}_{\substack{U \sim \omega' \\ W_{\overline{Q}} \sim V^{\overline{Q}}}} \left(\prod_{i \in \overline{Q}} \gamma_{u_i, w_i} \right) (p_1 + p_2)^{|Q|} \frac{1}{\Delta} \sum_{\substack{i,j \in Q \\ i \neq j}} \mathop{\mathbf{E}}_{(V_1 \cup \ldots \cup V_t) \sim \mathcal{V}} \sum_{t' \in [t]} \mathbf{1}[u_i, u_j \in V_{t'}]$$

$$\geq \mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{D}) - (p_1 + p_2) \frac{1}{\Delta} \left(\frac{2k}{\sqrt{n}} + \frac{k^2}{\Delta' \sqrt{n}} \right) \geq \mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{D}) - (p_1 + p_2) \frac{2k^2}{\Delta \Delta' \sqrt{n}}.$$

(by well-spreadness of ω' and the maximum size of $|V_t|$)

In all, we have proved that $\mathbf{E}_{\mathcal{E} \sim \tilde{\mathcal{E}}} \mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{E}) \geq \mathsf{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{D}) - (p_1 + p_2) \frac{2k^2}{\Delta \Delta' \sqrt{n}}$. Since $\tilde{\mathcal{E}}$ can be sampled in polynomial time, there is a randomized polynomial-time algorithm to find out a $\mathcal{E}$ satisfying (4.13). □

4.8 Bisection MAXkCSP

In this section, we consider the bisection MAX-CSP as a notable example of MAX-CSP with globally cardinality constraints.

yFix a finite domain D and a kCSPinstance $\mathcal{I}$ over D. A global cardinality constraint is a linear constraint on the numbers of variables that are assigned to the values in D. For simplicity and illustration purpose, here we only consider the bisection constraint – i.e., assuming $D = \{0, 1\}$, the number of variables that take value 1 is exactly $n/2$ (for even integers n). For a bisection kCSPinstance $\mathcal{I} = (V, \omega, P)$, we define its *optimal value* to be

$$\mathsf{opt}(\mathcal{I}) = \max_{\alpha : |\{v \in V : \alpha(v) = 1\}| = n/2} \{\mathsf{val}(\mathcal{I}, \alpha)\},$$

where the definition of $\mathsf{val}(\mathcal{I}, \alpha)$ remains the same as in the ordinary kCSPcase.

The ℓ-round Sherali-Adams relaxation for a bisection kCSPinstance $\mathcal{I} = (V, \omega, P)$

($\ell \geq k$) is written as follows.

$$
\begin{aligned}
\text{maximize} \quad & \mathop{\mathbf{E}}_{S\sim\omega} \mathop{\mathbf{E}}_{\alpha\sim\boldsymbol{\mu}_S}[P_S(\alpha)] \\
\text{subject to} \quad & \mathop{\Pr}_{\alpha\sim\boldsymbol{\mu}_S}[\alpha|_{S\cap T} = \beta] = \mathop{\Pr}_{\alpha\sim\boldsymbol{\mu}_T}[\alpha|_{S\cap T} = \beta] \\
& \qquad\qquad\qquad\qquad\qquad \forall S, T \subseteq V, |S \cup T| \leq \ell, \beta \in D^{S\cap T} \\
& \sum_{v\in V} \mathop{\Pr}_{\alpha\sim\boldsymbol{\mu}_{S\cup\{v\}}}[\alpha|_S = \beta \text{ and } \alpha(v) = 1] = \frac{n}{2} \cdot \boldsymbol{\mu}_S(\beta) \\
& \qquad\qquad\qquad\qquad\qquad \forall S \subseteq V, S < \ell, \beta \in \{0,1\}^S,
\end{aligned}
$$

where the last constraint corresponds to the bisection constraint.

We now turn to how to round κ-independent solutions. The following lemma is similar to Lemma 4.4.1.

Lemma 4.8.1. *Let $\mathcal{I} = (V, \omega, P)$ be a bisection kCSPinstance. Let $\boldsymbol{\mu}$ be an κ-independent solution (with respect to both uniform distribution and ω, $0 \leq \kappa \leq 1$) to the k-round Sherali-Adams LP relaxation. There is a randomized polynomial time algorithm to find an assignment $\alpha : V \to \{0,1\}$ such that $\mathsf{val}(\mathcal{I}, \alpha) \geq \mathsf{val}^{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}) - 3k\kappa^{1/4}$ and $|\{v \in V : \alpha(v) = 1\}| = n/2$.*

Proof. We sample α in the same way as we did in the proof of Lemma 4.4.1, and we see that $\mathbf{E}_\alpha[\mathsf{val}(\mathcal{I}, \alpha)] \geq \mathsf{val}_{\mathrm{LP}}(\mathcal{I}) - 2\sqrt{\kappa}$. Also observe that

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{\alpha}\left|\sum_{v\in V}\alpha(v) - \frac{n}{2}\right| \leq \sqrt{\mathop{\mathbf{E}}_{\alpha}\left(\sum_{v\in V}\alpha(v) - \frac{n}{2}\right)^2} \\
&= \sqrt{\sum_{v_1,v_2\in V}\mathop{\mathbf{E}}_{\alpha}[\alpha(v_1)\alpha(v_2)] - n\sum_{v\in V}\mathop{\mathbf{E}}_{\alpha}[\alpha(v)] + \frac{n^2}{4}} = \sqrt{\sum_{v_1,v_2\in V}\mathop{\mathbf{E}}_{\alpha}[\alpha(v_1)\alpha(v_2)] - \frac{n^2}{4}} \\
&\qquad\qquad \leq \sqrt{\sum_{v_1,v_2\in V}\mathop{\Pr}_{\beta\sim\boldsymbol{\mu}_{\{v_1,v_2\}}}[\beta(v_1) = \beta(v_2) = 1] + \sqrt{\kappa} - \frac{n^2}{4}} = \kappa^{1/4},
\end{aligned}
$$

where the last inequality is because of κ-independence with respect to uniform distribution, the definition of total correlation, and Lemma 4.2.1; the last equality is because of Sherali-Adams constraints.

In all, we have

$$
\mathop{\mathbf{E}}_{\alpha}\left(\mathsf{val}(\mathcal{I}, \alpha) - k\left|\sum_{v\in V}\alpha(v) - \frac{n}{2}\right|\right) \geq \mathsf{val}^{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}) - 2\sqrt{\kappa} - k\kappa^{1/4}
$$

$$\geq \mathsf{val}^{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}) - 2k\kappa^{1/4}.$$

We can sample an α in expected polynomial time so that

$$\mathop{\mathbf{E}}_{\alpha}\left[\mathsf{val}(\mathcal{I}, \alpha) - \left|\sum_{v \in V} \alpha(v) - \frac{n}{2}\right|\right] \geq \mathsf{val}_{\mathrm{LP}}(\mathcal{I}) - 3k\kappa^{1/4}.$$

By greedily rearranging $\left|\mathbf{E}_{v\in V}[\alpha(v)] - \frac{1}{2}\right|$-fraction of the entries in α, we get a bisection assignment α' such that $\mathsf{val}(\mathcal{I}, \alpha') \geq \mathsf{val}^{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}) - 3k\kappa^{1/4}$. □

Finally as a counterpart to Theorem 4.5.1, we show the following.

Theorem 4.8.2. *Let $\mathcal{I} = (V, \omega, P)$ be a bisection kCSPinstance over domain $\{0, 1\}$ such that ω is Δ-dense or Δ-locally dense. For any $\epsilon > 0$, let $\ell = \frac{2\cdot 3^4 k^6 4^k \log |D|}{\epsilon^4 \Delta}$. The additive integrality gaps of the $(\ell+k)$-round Sherali-Adams LP relaxation is at most ϵ; and there is a randomized rounding algorithm producing a solution whose value is at least $\mathsf{opt}(\mathcal{I}) - \epsilon$, in expected $n^{O(\ell)}$ time.*

Proof. Let $\boldsymbol{\mu}$ be a solution to the $(\ell+k)$-round Sherali Adams LP relaxation. Similar as in the proof of Theorem 4.5.1, Corollary 4.3.4 and Corollary 4.3.6, we know that there exists $t \leq \ell$ such that

$$\begin{aligned}
&\mathop{\mathbf{E}}_{U\sim V^t} \sqrt[4]{\mathop{\mathbf{E}}_{S\sim V^k} C_{\boldsymbol{\mu}}(\boldsymbol{x}_S|\boldsymbol{x}_U) + \mathop{\mathbf{E}}_{S\sim\omega} C_{\boldsymbol{\mu}}(\boldsymbol{x}_S|\boldsymbol{x}_U)} \\
\leq &\sqrt[4]{\mathop{\mathbf{E}}_{U\sim V^t}\left(\mathop{\mathbf{E}}_{S\sim V^k} C_{\boldsymbol{\mu}}(\boldsymbol{x}_S|\boldsymbol{x}_U) + \mathop{\mathbf{E}}_{S\sim\omega} C_{\boldsymbol{\mu}}(\boldsymbol{x}_S|\boldsymbol{x}_U)\right)} \\
\leq &\sqrt[4]{2\cdot\frac{k^2 4^k \log|D|}{\Delta\ell}} \leq \frac{\epsilon}{3k^2}.
\end{aligned}$$

Therefore, together with Lemma 4.3.2, we have

$$\mathop{\mathbf{E}}_{U\sim V^t}\left(\mathsf{val}_{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}|\boldsymbol{x}_U) - 3k\left(\sqrt[4]{\mathop{\mathbf{E}}_{S\sim V^k} C_{\boldsymbol{\mu}}(\boldsymbol{x}_S|\boldsymbol{x}_U)} + \sqrt[4]{\mathop{\mathbf{E}}_{S\sim\omega} C_{\boldsymbol{\mu}}(\boldsymbol{x}_S|\boldsymbol{x}_U)}\right)\right) \\ \geq \mathsf{val}_{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}) - \epsilon.$$

We enumerate all the possible ways of conditioning, and find out a solution $\boldsymbol{\mu}'$ to the $(k + \ell - t)$-round Sherali-Adams LP relaxation such that

$$\mathsf{val}_{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}') - 3k\left(\sqrt[4]{\mathop{\mathbf{E}}_{S\sim V^k} C_{\boldsymbol{\mu}'}(\boldsymbol{x}_S)} + \sqrt[4]{\mathop{\mathbf{E}}_{S\sim\omega} C_{\boldsymbol{\mu}'}(\boldsymbol{x}_S)}\right) \geq \mathsf{val}_{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}) - \epsilon.$$

Since $\boldsymbol{\mu}'$ is always κ-independent with respect to both uniform distribution and ω for $\kappa = \mathbf{E}_{S\sim V^k} C_{\boldsymbol{\mu}'}(\boldsymbol{x}_S) + \mathbf{E}_{S\sim\omega} C_{\boldsymbol{\mu}'}(\boldsymbol{x}_S)$, by Lemma 4.8.1, given $\boldsymbol{\mu}'$, we can find an assigment with value at least $\mathsf{val}_{\mathrm{LP}}(\mathcal{I}, \boldsymbol{\mu}) - \epsilon$ in randomized polynomial time. □

4.9 A gap instance for relaxation $\mathcal{H}$

In this section, we show a gap instance for the relaxation $\mathcal{H}$. Consider the following 2-AP instance $\mathcal{I}([5], \omega)$. Let us define $\omega_{i,j,p,q} = \frac{1}{64} A_{i,j} B_{p,q}$, where

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad B = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

If we view A and B as the adjacency matrices of two 5-vertex graphs, $\mathsf{val}(\mathcal{I}, \pi)$ is the number of edges in A that are mapped to an edge in B by π, multiplied by $\frac{25}{32}$. Since A is a 4-cycle with one isolated vertex, and B is a 3-cycle plus an edge, at most 2 edges in A can be mapped to B. Therefore, $\mathsf{opt}(\mathcal{I}) = \frac{25}{16}$.

On the other hand, let us consider the following distribution $\mathcal{D}$ of permutations, where $\mathcal{D}$ is supported on π_1 and π_2 with equal probability $(1/2)$. π_1 is the identity permutation; $\pi_2(i) = (i \bmod 5) + 1$ for all $i \in [5]$. We have

$$\begin{aligned}
&\mathsf{val}(\mathcal{I}, \mathcal{D}) \\
=&\frac{25}{64} \cdot \sum_{i,j} \sum_{p,q} A_{i,j} B_{p,q} \frac{1}{2} \left(\mathbf{1}[\pi_1(i) = p \text{ or } \pi_2(i) = p]\right) \cdot \frac{1}{2} \left(\mathbf{1}[\pi_1(j) = q \text{ or } \pi_2(j) = q]\right) \\
=&\frac{225}{128} > \mathsf{opt}(\mathcal{I}).
\end{aligned}$$

Chapter 5

Lasserre integrality gaps for DENSEkSUBGRAPH

5.1 Introduction

As we defined at the beginning of this thesis, the DENSEkSUBGRAPH problem takes as input a graph $G(V, E)$ on n vertices and a parameter k, and asks for a subgraph of G on at most k vertices having the maximum number of edges.

While it is a fundamental graph optimization problem and arises in several applications (community detection in social networks, identifying protein families and molecular complexes in protein-protein interaction networks, etc), there is a huge gap between the best approximation algorithm and the known inapproximability results. The current best approximation algorithm due to [41] gives $O(n^{1/4+\epsilon})$-factor approximation algorithm which runs in time $n^{O(1/\epsilon)}$ for any constant $\epsilon > 0$. On the inapproximability side, [88] initially showed a small constant factor inapproximability for DENSEkSUBGRAPH using the random 3-SAT assumption. [137] used quasi-random PCPs to rule out a PTAS. More recently, [193, 7] used more non-standard assumptions to rule out any constant factor approximation algorithms.

While only constant factor approximations have been ruled out, it is commonly believed that DENSEkSUBGRAPH is much harder to approximate even on average (for a natural distribution on hard instances). Recently, average-case hardness assumptions based on the hardness of "planted" versions of DENSEkSUBGRAPH were used for public key cryptography [12] and in showing that financial derivates can be fraudulently priced without detection [15]. Given the interest in DENSEkSUBGRAPH from both the algorithms and the complexity point of view, developing a better understanding of the problem is an important challenge for the field.

5.1.1 Our contributions

In this chapter, we study the limitation of the most powerful Lasserre SDP relaxation hierarchy for DENSEkSUBGRAPH. We show an integrality gap of polynomial ratio (n^ϵ, for small enough constant ϵ) for almost linear ($n^{O(\epsilon)}$) levels of the Lasserre relaxation. If we only aim at an integrality gap for polynomial (n^ϵ) levels of the Lasserre relaxation, the ratio of the gap can be as large as $n^{2/53-O(\epsilon)}$. Informally, we prove

Theorem 5.1.1. *[Informal version of Theorem 5.3.6 and Theorem 5.3.7] For every $\epsilon > 0$, there is a lower bound of $n^{2/53-\epsilon}$ on the integrality gap of level-$n^{\Omega(\epsilon)}$ Parrilo-Lasserre SDP relaxation hierarchy for the* DENSEkSUBGRAPH *problem; there is also a lower bound of n^ϵ on the integrality gap of level-$n^{1-O(\epsilon)}$ Parrilo-Lasserre SDP relaxation hierarchy.*

As we mentioned in the introduction part of this thesis, our integrality gaps are among the few known cases where the integrality gap ratio is much bigger than the best known NP-hardness inapproximability bounds. In the absence of inapproximability results for DENSEkSUBGRAPH, our results show that beating a factor of $n^{\Omega(1)}$ is a barrier for even the most powerful SDPs, and in fact even beating the best known $n^{1/4}$ factor is a barrier for current techniques.

5.2 Preliminaries

5.2.1 Notations

We introduce some notation which will be used throughout this chapter. $G = (V, E)$ refers to a graph which is an instance of the DENSEkSUBGRAPH problem on n vertices, and k refers to the size of the subgraph we are required to output. For an induced subgraph $H \subseteq G$, we denote by $d(H)$ the average degree (or *density* of H). For a vertex v in subgraph H, we will denote by $\Gamma_H(v)$ the set of neighbors of v in H (the suffix will be dropped when $H = G$).

The phrase "with high probability" will mean: with probability $1 - \frac{1}{p(n)}$, for any polynomial $p(n)$. It will be clear from the context that there are constants which depend on the degree of p.

5.2.2 The natural and min degree integer programmings for DENSEkSUBGRAPH

The natural integer programming for DENSEkSUBGRAPH has variables $\{x_i\}$ to denote if vertex i belongs to the solution, and edge variables $\{x_{ij}\}_{(i,j)\in E(G)}$ to denote if both i, j are in the subgraph.

For our integrality gaps, we will also consider to start with a different integer programming (DkS-IP2) which is equivalent upto a factor of 2 (see [41]). Intuitively, it tries to find a k-subgraph H such that the *minimum degree* d_H is maximized. An LP hierarchy obtained from this min. degree IP (DkS-IP2) was in fact used by [41] to obtain their approximation algorithm. (While the program as stated is not linear, we guess the degree d and consider the feasibility linear program that is obtained.)

Natural IP (DkS-IP1)	**Min degree IP (DkS-IP2)**
$$\begin{aligned} \text{Maximize} \quad & \sum_{(i,j)\in E(G)} x_i x_j \\ \text{s.t.} \quad & \sum_{i\in V} x_i \le k \\ & \forall i \in V \quad x_i \in \{0,1\} \end{aligned}$$	$$\begin{aligned} \text{Maximize} \quad & d \\ \text{s.t.} \quad & \sum_{i\in V} x_i \le k, \\ & \forall i \in V \quad \sum_{j\in\Gamma(i)} x_i x_j \ge d x_i \\ & \forall i, j \in V \quad x_{ij} = x_{ji} \\ & \forall i \in V \quad x_i \in \{0,1\} \end{aligned}$$

5.2.3 The Lasserre hierarchy for DENSEkSUBGRAPH

As in [66], the r-level Lasserre SDP for DENSEkSUBGRAPH for the natural IP (DkS-IP1) introduces a vector $\boldsymbol{U}_S$ for each subset $S \subseteq V$ with $|S| \le r$.

Lasserre hierarchy (r levels) for DENSEkSUBGRAPH:

$$\sum_{(u,v)\in E} \left\|\boldsymbol{U}_{\{u,v\}}\right\|^2$$

such that

$$\langle \boldsymbol{U}_{S_1}, \boldsymbol{U}_{S_2}\rangle \geq 0 \text{ for all } S_1, S_2$$
$$\langle \boldsymbol{U}_{S_1}, \boldsymbol{U}_{S_2}\rangle = \langle \boldsymbol{U}_{S_3}, \boldsymbol{U}_{S_4}\rangle \text{ when } S_1 \cup S_2 = S_3 \cup S_4$$
$$\sum_{v\in V} \left\|\boldsymbol{U}_{\{v\}}\right\|^2 \leq k$$
$$\left\|\boldsymbol{U}_{\emptyset}\right\|^2 = 1$$

The intended solution sets $\boldsymbol{U}_S = \boldsymbol{U}_\emptyset$ if every vertex in S belongs to the densest k-subgraph, and $\boldsymbol{U}_S = \boldsymbol{0}$ otherwise. The vector lengths $\left\|\boldsymbol{U}_S\right\|^2$ correspond to valid LP values x_S for the Sherali-Adams relaxation presented above.

The Lasserre SDP for the min degree IP (DkS-IP2) tries to find the k-subgraph of largest induced minimum degree d. This can be captured by the SDP constraint

$$\forall u \in V, \quad \sum_{v\in\Gamma(u)} \left\|\boldsymbol{U}_{u,v}\right\|^2 \geq d \cdot \left\|\boldsymbol{U}_u\right\|^2 \tag{5.1}$$

We will show in Section 5.3.3.1 that our integrality gaps also hold for the Lasserre hierarchy defined by this SDP. We refer to the SDP with constraint (5.1) as the *Min degree Lasserre SDP* .

5.3 The integrality gap

In this section, we show a gap instance with arbitrary large constant ratio for linear-round Lasserre relaxation, and a gap instance with n^ϵ ratio for $n^{1-O(\epsilon)}$-round Lasserre relaxation (Theorem 5.3.6). We also aim at maximizing the ratio of a polynomial-round Lasserre gap instance, getting a ratio of $\Omega(n^{2/53-\epsilon})$ (Theorem 5.3.7).

Our construction is based on a variant of Tulsiani's gap instance for kCSP [220] – we extend the parameter range of Tulsiani's instance. Then we convert the kCSP instance to a constraint-variable graph and duplicate the variable vertices, which is our gap instance for DENSEkSUBGRAPH. Note that the gap for kCSP problem is indeed a set of random instances. The vector solution from Lasserre gap for kCSP will help us exhibit a good

Lasserre vector solution for DENSEkSUBGRAPH. We finally use the structure of random instances of kCSP to show the soundness holds with high probability.

Now, let us proceed to the first step, the gap instance for kCSP.

5.3.1 Lasserre gap for kCSP from Tulsiani

We start by defining the kCSP problem.

Definition 5.3.1. *Let $C \subseteq \mathbb{F}_q^K$ be a q-ary linear code of block length K.*

1. *An instance Φ of $k\mathrm{CSP}(C)$ is a set of constraints $C_1, C_2, \cdots, C_m$ where each constraint C_i is over a K-tuple $T_i = (x_{i_1}, x_{i_2}, \cdots, x_{i_K})$, and is of the form $(x_{i_1} + b_1^{(i)}, x_{i_2} + b_2^{(i)}, \cdots, x_{i_K} + b_K^{(i)}) \in C$ for some $b^{(i)} \in \mathbb{F}_q^K$.*
2. *A random instance of $k\mathrm{CSP}(C)$ is sampled by choosing each constraint C_i independently, where we sample K variables without replacement from $[n]$ to get $T_i = (x_{i_1}, x_{i_2}, \cdots, x_{i_K})$ and $b^{(i)}$ is chosen from $\mathbb{F}_q^K$ uniformly.*

The following theorem is an extension of the main theorem in [220], showing that polynomial-round Lasserre relaxation cannot refute random kCSP with high probability.

Theorem 5.3.2. *If C is the dual code of a distance $2\delta \geq 3$ code (in terms of number of coordinates, not fractional distance), for every $10 \leq K < n^{1/2}$, if $n^{\kappa-1} \leq \eta \leq 1/(10^8 \cdot (\beta K^{2\delta+0.75})^{1/(\delta-1)})$ for some $\kappa > 0$, then for large enough n, a random instance Φ of $k\mathrm{CSP}(C)$ over $m = \beta n$ constraints and n variables, with probability $1 - o(1)$, admits perfect solution for the SDP relaxation obtained by $\eta n/16$ rounds of the Lasserre hierarchy, i.e. there are vectors $\boldsymbol{V}_{(S,\alpha)}$ for all $S \subseteq [n]$ with $|S| \leq \eta n/16$ and all $\alpha : S \to \mathbb{F}_q$, such that*

- *the value of the solution is perfect: $\sum_{i=1}^m \sum_{\alpha:T_i\to\mathbb{F}_q} C_i(\alpha) \left\|\boldsymbol{V}_{T_i,\alpha}\right\|^2 = m$;*
- *$\langle \boldsymbol{V}_{(S_1,\alpha_1)}, \boldsymbol{V}_{(S_2,\alpha_2)}\rangle \geq 0$ for all $S_1, S_2, \alpha_1, \alpha_2$;*
- *$\langle \boldsymbol{V}_{(S_1,\alpha_1)}, \boldsymbol{V}_{(S_2,\alpha_2)}\rangle = 0$ if $\alpha_1(S_1 \cap S_2) \neq \alpha_2(S_1 \cap S_2)$;*
- *$\langle \boldsymbol{V}_{(S_1,\alpha_1)}, \boldsymbol{V}_{(S_2,\alpha_2)}\rangle = \langle \boldsymbol{V}_{(S_3,\alpha_3)}, \boldsymbol{V}_{(S_4,\alpha_4)}\rangle$ for all $S_1 \cup S_2 = S_3 \cup S_4$ and $\alpha_1 \circ \alpha_2 = \alpha_3 \circ \alpha_4$;*
- *$\left\|\boldsymbol{V}_{(\emptyset,\emptyset)}\right\|^2 = 1$ and $\sum_{j\in\mathbb{F}_q} \left\|\boldsymbol{V}_{(\{i\},\{i\to j\})}\right\|^2 = 1$ for all $i \in [n]$.*

Note that Theorem 5.3.2 extends the original theorem of [220] to the regime where K might be superconstant (even $\mathrm{poly}(n)$). The proof of Theorem 5.3.2 follows the proof in Tulsiani's paper, with the following changes.

Recall that Tulsiani showed that, if the constraint-variable graph of a $k\mathrm{CSP}(C)$ instance has very high left-expansion, then the Lasserre SDP admits perfect solution for it. Formally, the following lemma is (implicitly) shown in [220].

Lemma 5.3.3 ([220])**.** *Given a $k\mathrm{CSP}(C)$ instance, if every set of constraints of cardinality $s \le r$ involves more than $(K-\delta)s$ variables (where 2δ is the distance of the dual code of C), and if $4\delta \le K$, then there is a perfect solution for the SDP relaxation obtained by $r/16$ rounds of the Lasserre hierarchy.*

Hence, we only need to prove the following lemma which shows that the constraint-variable graph still has very high left-expansion, even when a constraint might involve superconstant many variables (i.e. the left degree might be superconstant).

Lemma 5.3.4. *Given β, η, K as in Theorem 5.3.2, with probability $1-o(1)$, for all $2 \le s \le \eta n$, every set of s constraints involves more than $(K-\delta)s$ variables.*

A similar lemma can be found in [220] (Lemma A.1), which only deals with constant K. We need a more refined argument for superconstant K, which is in Section 5.3.4.

5.3.2 The gap instance for DENSEkSUBGRAPH

The gap instance is reduced from the gap instance for kCSP in Theorem 5.3.2. Let C be the dual code of a $[K, K-t, 2\delta]_q$ code as used in Theorem 5.3.2, where K is the block length, $(K-t)$ is the dimension, and $2\delta \ge 3$ is the distance of the code. Such a code has size $|C| = q^t$, and is very sparse for small enough t. For $1000 < q$ and $K > q^2$, we let $\beta = (40q^{t+2}\ln q)/K$, and do the following reduction.

Given a $k\mathrm{CSP}(C)$ instance Φ with $m = \beta n$ constraints and n variables. Let $G_\Phi = (L_\Phi, R_\Phi, E_\Phi)$ be the bipartite graph with $m|C|$ left vertices and nq right vertices. For every constraint C_i and every partial assignment to variables in the corresponding tuple T_i which satisfies the constraint C_i, we introduce a left vertex. For every variable x_i and its corresponding assignment, we introduce a right vertex. Formally,

$$\begin{aligned} L_\Phi &= \{(C_i, \alpha) | i \in [m], \alpha : T_i \to \mathbb{F}_q, C_i(\alpha) = 1\}, \\ R_\Phi &= \{(x_j, \alpha) | j \in [n], \alpha : \{x_j\} \to \mathbb{F}_q\}. \end{aligned}$$

We connect a left vertex (C_i, α) and right vertex (x_j, α') when $x_j \in T_i$ and α' is consistent with α, i.e.

$$E_\Phi \;=\; \{\{(C_i, \alpha), (x_j, \alpha')\} | (C_i, \alpha) \in L_\Phi, x_j \in T_i, \alpha'(x_j) = \alpha(x_j)\}.$$

Now we define the final graph $G'_\Phi = (L_\Phi, R'_\Phi, E'_\Phi)$ in which we want to find a dense k-subgraph where $k = 2m$. We take β copies of the right vertices in R_Φ to get R'_Φ. To get E'_Φ, we connect a left vertex $u \in L_\Phi$ and a right vertex $v \in R'_\Phi$ if u is connected to v's corresponding vertex in R_Φ in E_Φ. The graph G'_Φ has $N = m|C| + \beta nq = O(nq^{2t+2} \ln q/K)$ vertices.

In our analysis of the reduction, we need a q-ary linear code $\mathcal{C}$ that has a small constant distance (but no less than 3), small block length (but more than q), and very high dimension. Thus, we instantiate the code $\mathcal{C}$ with Generalized BCH codes given by the following.

Lemma 5.3.5 (Generalized BCH Codes)**.** *For every prime tower q, and integer $2\delta \geq 3$, there are q-ary linear codes of block length $K = q^2 - 1$, dimension $(K - 4\delta + 3)$, and distance at least 2δ.*

We include a simple proof of Lemma 5.3.5 as follows.

Proof. Let γ be a primitive element of $\mathbb{F}_{q^2}$. Let $D = 2\delta$ for notational ease. We construct the following code

$$\tilde{C} = \{(c_1, c_2, \cdots, c_{q^2-1}) \in \mathbb{F}_q^{q^2-1} | c(1) = c(\gamma) = c(\gamma^2) = \cdots = c(\gamma^{D-2}) = 0, \text{ where } c(X) = c_1 X + c_2 X^2 + c_3 X^3 + \cdots + c_{q^2-1} X^{q^2-1}\}.$$

We first show that the distance of $\tilde{C}$ is at least D. Since $\tilde{C}$ is a linear code, we only need to show that every non-zero codeword has weight at least D.

We show the contrapositive statement : the only codeword of weight at most $D - 1$ is $\mathbf{0}$. For every codeword of weight at most $D - 1$, suppose the non-zero entries are in the set $\{c_{i_1}, c_{i_2}, c_{i_3}, \cdots, c_{i_{D-1}}\}$. We have

$$\begin{array}{rcrcrcccrl}
c_{i_1} & + & c_{i_2} & + & c_{i_3} & + & \cdots & + & c_{i_{D-1}} & =0 \\
\gamma^{i_1} c_{i_1} & + & \gamma^{i_2} c_{i_2} & + & \gamma^{i_3} c_{i_3} & + & \cdots & + & \gamma^{i_{D-1}} c_{i_{D-1}} & =0 \\
\gamma^{2i_1} c_{i_1} & + & \gamma^{2i_2} c_{i_2} & + & \gamma^{2i_3} c_{i_3} & + & \cdots & + & \gamma^{2i_{D-1}} c_{i_{D-1}} & =0 \\
 & & & & \vdots & & & & &
\end{array}$$

$$\gamma^{(D-2)i_1}c_{i_1} \quad + \quad \gamma^{(D-2)i_2}c_{i_2} \quad + \quad \gamma^{(D-2)i_3}c_{i_3} \quad + \quad \cdots \quad + \quad \gamma^{(D-2)i_{D-1}}c_{i_{D-1}} \quad =0$$

Note that the coefficients form a Vandermonde matrix (which has full rank). Therefore we have $c_{i_1} = c_{i_2} = c_{i_3} = \cdots = c_{i_{D-1}} = 0$, i.e. the codeword is $\mathbf{0}$.

Now we show that the dimension of $\tilde{C}$ is at least $(K - 2D + 3)$. Note that each constraint $c(\gamma^i) = 0 (1 \leq i \leq D - 2)$ can be implemented by 2 linear constraints in $\mathbb{F}_q$ (since $\gamma^i \in \mathbb{F}_{q^2}$), while the constraint $c(1) = 0$ is indeed a linear constraint in $\mathbb{F}_q$. Therefore, we need at most $2(D - 2) + 1 = 2D - 3$ linear constraints for $\tilde{C}$, i.e. the dimension of $\tilde{C}$ is at least $(K - 2D + 3)$.

Finally, if the dimension of $\tilde{C}$ is more than $(K-2D+3)$, we can take a linear subspace of $\tilde{C}$ of dimension $(K - 2D + 3)$, while the distance of the subspace code is no less than the distance of $\tilde{C}$. $\square$

5.3.3 Analysis

We get a family of gap instances G'_Φ parameterized by $q > 1000$ and $2\delta \geq 3$ (using Lemma 5.3.5). We obtain our two main results of this section by picking appropriate parameters for code C as follows. To get lasserre integrality gaps for $N^{1-O(\epsilon)}$ levels , we show the following by setting the distance $2\delta = 3$.

Theorem 5.3.6. *For every $1000 < q < N^\epsilon$ (where ϵ is an absolute small constant), there is a gap instance of ratio $\Omega(q)$ for $N/q^{O(1)}$-level Lasserre SDP. The same construction also works for the min degree Lasserre SDP, when $q = \Omega(\log n)$ and $q < N^\epsilon$.*

We now aim at getting a gap instance of ratio N^ϵ for polynomial-round Lasserre SDP, where ϵ is maximized. By setting $q = n^\gamma$ for some small constant $\gamma > 0$, the distance $2\delta = 4$, and optimizing the other parameters, we obtain the following (refer to section 5.3.3.4 for details)

Theorem 5.3.7. *For small enough $\kappa > 0$, there is a gap instance of ratio $N^{2/53-O(\kappa)}$ for the N^κ-round Min degree Lasserre SDP.*

The two theorems follow because of Theorem 5.3.2, Lemma 5.3.8, Lemma 5.3.9 (completeness) and Lemma 5.3.10 (soundness). In the completeness case, we will use our r-level Lasserre solution for kCSP to show that the Lasserre SDP after $R = r/K$ levels of the hierarchy has value at least βmK. In the soundness case, we show that with probability $1 - o(1)$, the graph G'_Φ does not have any $2m$-subgraph of value more than $17/q$ times the SDP value (Lemma 5.3.10). Therefore, the graph G'_Φ is a gap instance of ratio $\Omega(q)$ for R-round Lasserre SDP. We proceed by first proving these lemmas.

5.3.3.1 Completeness

Lemma 5.3.8. *If the kCSP(C) instance Φ admits perfect solution for r-round Lasserre SDP relaxation, then the r/K-round Lasserre SDP relaxation for the* DENSEkSUBGRAPH *instance G'_Φ has a solution of value βmK.*

Proof. For any set $S = L_\Phi \cup R'_\Phi$, suppose the left vertices included in S are

$$(C_{i_1}, \alpha_1), (C_{i_2}, \alpha_2), \cdots, (C_{i_{r_1}}, \alpha_{r_1}),$$

and the right vertices included in S are

$$(x_{j_1}, \alpha'_1), (x_{j_2}, \alpha'_2), \cdots, (x_{j_{r_2}}, \alpha'_{r_2}),$$

where $r_1 + r_2 \leq r/K$. Let

$$S' = T_{i_1} \cup T_{i_2} \cup \cdots \cup T_{i_{r_1}} \cup \{x_{j_1}\} \cup \{x_{j_2}\} \cup \cdots \cup \{x_{j_{r_2}}\}.$$

We have $|S'| \leq Kr_1 + r_2 \leq r$. If all the partial assignments α_i's and α'_i's are consistent to each other (i.e. there are not two of them assigning the same variable to different values), we can define

$$\alpha = \alpha_1 \circ \alpha_2 \circ \cdots \alpha_{r_1} \circ \alpha'_1 \circ \alpha'_2 \circ \cdots \alpha'_{r_2},$$

and let $\boldsymbol{U}_S = \boldsymbol{V}_{S',\alpha}$, or we let $\boldsymbol{U}_S = \mathbf{0}$.

We can check that all the Lasserre constraints are satisfied.

- For two sets S_1, S_2, either at least one of the vectors $\boldsymbol{U}_{S_1}, \boldsymbol{U}_{S_2}$ is $\mathbf{0}$ (therefore their inner-product is 0), or $\boldsymbol{U}_{S_1} = \boldsymbol{V}_{S'_1,\alpha_1}, \boldsymbol{U}_{S_2} = \boldsymbol{V}_{S'_2,\alpha_2}$ for some $S'_1, S'_2, \alpha_1, \alpha_2$ and $\langle \boldsymbol{U}_{S_1}, \boldsymbol{U}_{S_2} \rangle = \langle \boldsymbol{V}_{S'_1,\alpha_1}, \boldsymbol{V}_{S'_2,\alpha_2} \rangle \geq 0$.
- For any S_1, S_2, S_3, S_4 such that $S_1 \cup S_2 = S_3 \cup S_4$, either the set of partial assignments in $S_1 \cup S_2 = S_3 \cup S_4$ are consistent to each other, in which case we have $\boldsymbol{U}_{S_1 \cup S_2} = \boldsymbol{U}_{S_3 \cup S_4} = \boldsymbol{V}_{S,\alpha}$ where S is the union of all the variables included in $S_1 \cup S_2$ and α is the concatenation of the partial assignments in $S_1 \cup S_2$; or we have $\boldsymbol{U}_{S_1 \cup S_2} = \boldsymbol{U}_{S_3 \cup S_4} = \mathbf{0}$.

- The "(fractional) number of chosen vertices"

$$\begin{aligned}
\sum_{v\in L_\Phi\cup R'_\Phi}\left\|\boldsymbol{U}_{\{v\}}\right\|^2 &= \sum_{(C_i,\alpha)\in L_\Phi}\left\|\boldsymbol{U}_{\{(C_i,\alpha)\}}\right\|^2+\sum_{(x_j,\alpha)\in R'_\Phi}\left\|\boldsymbol{U}_{\{x_j,\alpha\}}\right\|^2\\
&=\sum_{i=1}^m\sum_{\alpha:T_i\to\mathbb{F}_q}C_i(\alpha)\left\|\boldsymbol{U}_{\{(C_i,\alpha)\}}\right\|^2+\beta\sum_{j=1}^n\sum_{\alpha:\{x_j\}\to\mathbb{F}_q}\left\|\boldsymbol{U}_{\{(x_j,\alpha)\}}\right\|^2\\
&=\sum_{i=1}^m\sum_{\alpha:T_i\to\mathbb{F}_q}C_i(\alpha)\left\|\boldsymbol{V}_{(T_i,\alpha)}\right\|^2+\beta\sum_{j=1}^n\sum_{\alpha:\{x_j\}\to\mathbb{F}_q}\left\|\boldsymbol{V}_{(\{x_j\},\alpha)}\right\|^2\\
&=m+\beta n=2m,
\end{aligned}$$

  and $\left\|\boldsymbol{U}_\emptyset\right\|^2=\left\|\boldsymbol{V}_{(\emptyset,\emptyset)}\right\|^2=1$.

Now, we calculate the value of the solution

$$\begin{aligned}
&\sum_{(u,v)\in E'_\Phi}\left\|\boldsymbol{U}_{\{u,v\}}\right\|^2=\beta\sum_{(u,v)\in E_\Phi}\left\|\boldsymbol{U}_{\{u,v\}}\right\|^2=\beta\sum_{i=1}^m\sum_{\alpha:T_i\to\mathbb{F}_q,C_i(\alpha)=1}\sum_{x_j\in T_i}\left\|\boldsymbol{U}_{\{(C_i,\alpha),(x_j,\alpha_{|\{x_j\}})\}}\right\|^2\\
=&\beta\sum_{i=1}^m\sum_{\alpha:T_i\to\mathbb{F}_q,C_i(\alpha)=1}K\left\|\boldsymbol{V}_{(T_i,\alpha)}\right\|^2=\beta\sum_{i=1}^m K=\beta mK.
\end{aligned}$$

□

If we add the constraint (5.1), we can still get a good SDP solution for the Min degree Lasserre SDP with high probability, as long as q is superconstant.

Lemma 5.3.9. *For $q=\Omega(\log n)$, with probability $1-o(1)$, this vector solution also satisfies the added constraint* (5.1) *with $d=\beta K/2$, i.e., for each vertex u, we have*

$$\sum_{v\in\Gamma(u)}\left\|\boldsymbol{U}_{\{u,v\}}\right\|^2\geq\beta K/2\cdot\left\|\boldsymbol{U}_u\right\|^2.$$

Proof. For each left vertex (C_i,α), we have

$$\sum_{v\in\Gamma((C_i,\alpha))}\left\|\boldsymbol{U}_{\{(C_i,\alpha),v\}}\right\|^2=\beta\sum_{x_j\in T_i}\left\|\boldsymbol{U}_{\{(C_i,\alpha),(x_j,\alpha_{|\{x_j\}})\}}\right\|^2=\beta\sum_{x_j\in T_i}\left\|\boldsymbol{U}_{\{(C_i,\alpha)\}}\right\|^2=\beta K\left\|\boldsymbol{U}_{\{(C_i,\alpha)\}}\right\|^2.$$

For each right vertex (x_j,α'), we have

$$\sum_{v\in\Gamma((x_j,\alpha'))}\left\|\boldsymbol{U}_{\{(x_j,\alpha'),v\}}\right\|^2=\sum_{i:T_i\ni x_j}\sum_{\alpha:T_i\to\mathbb{F}_q,C_i(\alpha)=1,\alpha(x_j)=\alpha'(x_j)}\left\|\boldsymbol{U}_{\{(x_j,\alpha'),(C_i,\alpha)\}}\right\|^2$$

$$= \sum_{i:T_i \ni x_j} \sum_{\alpha:T_i \to \mathbb{F}_q, C_i(\alpha)=1, \alpha(x_j)=\alpha'(x_j)} \left\| \boldsymbol{U}_{\{(C_i,\alpha)\}} \right\|^2$$
$$= \sum_{i:T_i \ni x_j} \sum_{\alpha:T_i \to \mathbb{F}_q, \alpha(x_j)=\alpha'(x_j)} \left\| \boldsymbol{U}_{\{(C_i,\alpha)\}} \right\|^2,$$

where the last equality is because we know that $\boldsymbol{U}_{\{(C_i,\alpha)\}} = \mathbf{0}$ when $C_i(\alpha) \neq 1$. By the property of Lasserre vectors, we know that for each $i \in [m]$,

$$\sum_{\alpha:T_i \to \mathbb{F}_q, \alpha(x_j)=\alpha'(x_j)} \left\| \boldsymbol{U}_{\{(C_i,\alpha)\}} \right\|^2 = \left\| \boldsymbol{U}_{\{(x_j,\alpha')\}} \right\|^2,$$

therefore

$$\sum_{v \in \Gamma((x_j,\alpha'))} \left\| \boldsymbol{U}_{\{(x_j,\alpha'),v\}} \right\|^2 = \sum_{i:T_i \ni x_j} \left\| \boldsymbol{U}_{\{(x_j,\alpha')\}} \right\|^2.$$

For $q = \Omega(\log n)$, the expected number of constraints containing x_j is $\beta K = \Omega((\log n)^{t+2}) = \Omega(\log n)$, by our choice of β. Therefore, by Chernoff bound and union bound, with probability $1 - o(1)$, for all x_j, there are at least $\beta K/2$ constraints containing x_j, and we have

$$\sum_{v \in \Gamma((x_j,\alpha'))} \left\| \boldsymbol{U}_{\{(x_j,\alpha'),v\}} \right\|^2 \geq \beta K/2 \cdot \left\| \boldsymbol{U}_{\{(x_j,\alpha')\}} \right\|^2,$$

for every x_j and α'. □

5.3.3.2 Soundness

Now, we show that random instances of kCSP give rise to graphs G'_ϕ whose $2m$-sized subgraphs have density $O(\beta K/q)$. Note that the large alphabet size q allows us to get a much larger gap than we would starting from random AND instances [88]. This allows us some slack in the size of the subgraphs we need to argue about.

For C the dual of a $[K, K-t, 2\delta]_q$ code, we prove the following soundness lemma.

Lemma 5.3.10. *When $\beta \geq (40q^{t+2} \ln q)/K$, for a random kCSP(C) instance Φ, with probability $1 - o(1)$, any subgraph of G'_Φ obtained by choosing $2m$ left vertices and $2m$ right vertices contains at most $17\beta mK/q$ edges, and therefore any $2m$-subgraph of G'_Φ contains at most $17\beta mK/q$ edges.*

Note that G'_ϕ was constructed by taking β copies of the right bipartition and replicating the edges. To prove Lemma 5.3.10, we only need to prove the following lemma.

Lemma 5.3.11. *Suppose that $q > 1000, K > q^2/2, t \leq 10$. When $\beta \geq (40q^{t+2}\ln q)/K$, for a random kCSP(C) instance Φ, with probability $1-o(1)$, any subgraph of G_Φ obtained by choosing $2m$ left vertices and $2n$ right vertices contains at most $17mK/q$ edges.*

Proof of Lemma 5.3.10 from Lemma 5.3.11. We only need to prove once there is a $2m \times 2m$ subgraph of G'_Φ with t edges, there is a $2m \times 2n$ subgraph of G_Φ with at least t/β edges. Fix $2m$ left vertices in G'_Φ, to maximize the number of edges in the subgraph, we need to select the $2m$ right vertices with most edges connected to the chosen $2m$ left vertices. Since any two right vertices G'_Φ corresponding to the same right vertex in G_Φ have the same set of neighbors, there is an densest $2m \times 2m$ subgraph H' of G'_Φ that, for any two such vertices, chooses either both or neither of them. Now we define an subgraph H of G_Φ that contains the same $2m$ left vertices. It contains a right vertex if any copy of the vertex is contained in H'. H contains $2m/\beta = 2n$ vertices, and it is easy to see that there are (at least) t/β edges in H. □

We proceed by fixing a set of $2n$ vertices R on the right. Lemma 5.3.11 follows from the following claim by a standard union bound over all possible choices of R.

Claim 5.3.12. *Recall that $G_\Phi = (L_\Phi, R_\Phi, E_\Phi)$. Suppose that $q > 1000, K > q^2/2, t \leq 10$. Fix a subset $R \subseteq R_\Phi$ (note that R_Φ is the same for all the instances Φ of n variables), $|R| = 2n$, the probability (over choice of Φ) that there does not exist a subset $L \subseteq L_\Phi$ of size $2m$ such that the number of edges in the induced subgraph by $L \cup R$ is more than $17mK/q$, is at least $1 - \exp(-mK/(10q^{t+2}))$.*

Proof of Lemma 5.3.11 from Claim 5.3.12. Since there are only $\binom{qn}{2n} \leq \exp(2n(\ln q + 1))$ choices of R, by a union bound, with probability at least

$$1 - \exp(2n(\ln q + 1)) \cdot \exp(-mK/(10q^{t+2})) = 1 - \exp(2n(\ln q + 1) - \beta nK/(10q^{t+2})),$$

there is no $2m \times 2n$ subgraph of G_Φ containing more than $17mK/q$ edges. The probability becomes $1 - o(1)$ when $\beta = (40q^{t+2}\ln q)/K$. □

Proof of Claim 5.3.12. First, we show that with high probability, a constraint C_i is "poorly satisfied". That is, none of the left vertices corresponding to a constraint C_i has more than $\Omega(K/q)$ neighbors in R – this number is roughly $1/q$ times the corresponding value in completeness case. We prove this in the following two steps.

Step 1. Fix a subset $R \subseteq R_\Phi$, $|R| = 2n$, for each variable x_j, let $\deg(x_j)$ be the number of vertices in R that corresponding to x_j, i.e. let $\deg(x_j) = |R \cap \{(x_j, \alpha) | \alpha : \{x_j\} \to \mathbb{F}_q\}|$.

For a subset of variables $T \subseteq \{x_1, x_2, \cdots, x_n\}$, let $\deg(T) = \sum_{x_j \in T} \deg(x_j)$. We call T *good* if the average degree of variables in T is not more than 4, i.e. $\deg(T) \le 4|T|$.

For a random T with $|T| = K$, note that the expected degree $\mathbf{E}[\deg(T)] = 2K$. Therefore, by Hoeffding's inequalities for sampling without replacement (Theorem 1 and Theorem 4 in [119]), we have

$$\Pr[T \text{ is not good}] = \Pr[\deg(T) > 4K] < \exp(-\ln(4/e) \cdot 2K/q) < \exp(-K/(2q)).$$

Step 2. Again, fix $R \subseteq R_\Phi$, $T \subseteq \{x_1, x_2, \cdots, x_n\}$, for a codeword α on coordinates in T, i.e. $\alpha : T \to \mathbb{F}_q$, let $\mathrm{agr}_T(\alpha, R) = |\{(x_j, \alpha_{|x_j}) | x_j \in T\} \cap R|$. For a constraint C_i, say it is *poorly satisfied* if for all $\alpha : T \to \mathbb{F}_q$ such that $C_i(\alpha) = 1$, we have $\mathrm{agr}_{T_i}(\alpha, R) \le 8K/q$.

Recall that to sample a random constraint C_i, we first sample a random K-tuple T_i, and a random shifting function $b^{(i)}$. Note that for a fixed $\alpha : T \to \mathbb{F}_q$, and a fixed T_i that is good, when we take a random shifting function $b^{(i)} : T_i \to \mathbb{F}_q$, we have $\mathbf{E}_{b^{(i)}:T_i \to \mathbb{F}_q}[\mathrm{agr}_{T_i}(\alpha - b^{(i)}, R)] = \deg(T_i)/q = 4K/q$, therefore, by standard Chernoff bound, for a fixed codeword $\alpha \in C$, the probability that α makes C_i not poorly satisfied is bounded from above by

$$\Pr_{b^{(i)}}[\mathrm{agr}_{T_i}(\alpha - b^{(i)}, R) > 8K/q] < \exp(-\ln(4/e) \cdot 4K/q) < \exp(-K/q).$$

Since there are $|C| = q^t \le q^{10}$ codewords, by a union bound, for $K > q^2/2$ and $q > 1000$, we have

$$\Pr[C_i \text{ is not poorly satisfied} | T_i \text{ is good}] < q^{10} \cdot \exp(-K/q) < \exp(-K/(2q)).$$

In all, we have

$$\begin{aligned}\Pr[C_i \text{ is poorly satisfied}] \ge{}& \Pr[C_i \text{ is poorly satisfied} | T_i \text{ is good}] \cdot \Pr[T \text{ is good}] \\ &> (1 - \exp(-K/q))(1 - \exp(-K/(2q))) > 1 - \exp(-K/(3q)).\end{aligned}$$

Now, again, by standard Chernoff bound, we have

$$\begin{aligned}&\Pr[|\{C_i | C_i \text{ is not poorly satisfied}\}| > m/(q \cdot |C|)] < (e \cdot |C| \cdot q \cdot \exp(-K/(3q)))^{m/(q \cdot |C|)} \\ &\le \left(e \cdot q^{10} \cdot q \cdot \exp(-K/(3q))\right)^{m/(q \cdot |C|)} < \exp(-K/(10q))^{m/(q \cdot |C|)} = \exp(-mK/(10q^{t+2})).\end{aligned}$$

By the calculation above we know that with probability at least $1 - \exp(-mK/(10q^{t+2}))$, there are at most $m/(q \cdot |C|)$ constraints that are not poorly satisfied.

For each left vertex $(C_i, \alpha) \in L_\Phi$, if C_i is poorly satisfied, we know there are at most $8K/q$ edges from (C_i, α) to R. If C_i is not poorly satisfied, there are at most K edges to R_Φ – this upperbound also applies to R.

Therefore, with probability at least $1-\exp(-mK/(10q^{t+2}))$, any set of $2m$ left vertices has at most $2m \cdot 8K/q + m/(q \cdot |C|) \cdot |C| \cdot K \leq 17mK/q$ edges connected to R. □

We now complete the proofs of the main theorems in this section.

5.3.3.3 Proof of Theorem 5.3.6

By combining Theorem 5.3.2, Lemma 5.3.8, Lemma 5.3.9 (completeness), and Lemma 5.3.10 (soundness) we see that with probability $1-o(1)$, the graph G'_Φ provides a $\Omega(q)$ integrality for the number of levels R given by

$$R = \Omega\left(\frac{n}{K(\beta K^{2\delta+0.75})^{1/(\delta-1)}}\right) = \Omega\left(\frac{N}{q^{2t+2}\ln q \cdot (\beta K^{2\delta+0.75})^{1/(\delta-1)}}\right)$$
$$= \Omega\left(\frac{N}{K^{(2\delta-0.25)/(\delta-1)} q^{(2t+2)+(t+2)/(\delta-1)} \text{poly} \log q}\right). \quad \text{(recall that } \beta = (40q^{t+2}\ln q)/K\text{)}$$

Recall that $K = q^2 - 1$. By setting $K = q^2 - 1$ and $2\delta = 3$, we verify that the theorem holds. □

5.3.3.4 Proof of Theorem 5.3.7

Let $q = n^\gamma$, since $N = O(nq^{2t+2}\ln q/K) = O(nq^{2t}\ln q)$, ratio of the gap due to Lemma 5.3.8 and Lemma 5.3.10 is

$$\Omega(q) = \Omega(N^{\gamma/(1+2t\gamma+o(1))}) = \Omega(N^{\gamma/(1+(8\delta-6)\gamma+o(1))}). \qquad \text{(note that } t = 4\delta - 3\text{)}$$

This means that

$$\epsilon = \frac{\gamma}{1+(8\delta-6)\gamma+o(1)} = \frac{1}{8\delta-6} - \frac{1+o(1)}{(1+(8\delta-6)\gamma+o(1))(8\delta-6)}.$$

Note that when $2\delta \geq 3$ is fixed, ϵ is maximized when γ is maximized.

The number of rounds (due to Theorem 5.3.2 and Lemma 5.3.8) is

$$R = \Omega\left(\frac{N}{K^{(2\delta-0.25)/(\delta-1)} q^{(2t+2)+(t+2)/(\delta-1)} \text{poly} \log q}\right)$$

$$
\begin{aligned}
&= \Omega\left(\frac{N}{n^{2\gamma(2\delta-0.25)/(\delta-1)} \cdot n^{\gamma((8\delta-4)+(4\delta-1)/(\delta-1))+o(1)}}\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad (\text{by } K = \Theta(q^2),\, q = n^\gamma,\, t = 4\delta - 3) \\
&= \Omega\left(\frac{N}{n^{\gamma(8\delta+4+6.5/(\delta-1))+o(1)}}\right) \\
&= \Omega\left(N^{1-\frac{\gamma(8\delta+4+6.5/(\delta-1))+o(1)}{1+o(1)+\gamma(8\delta-6)}}\right)
\end{aligned}
$$

For very small $\kappa > 0$, to get a gap instance for $N^{\Omega(\kappa)}$-round Lasserre, we need

$$
\begin{aligned}
&1 - \frac{\gamma(8\delta+4+6.5/(\delta-1))+o(1)}{1+o(1)+\gamma(8\delta-6)} \geq \Omega(\kappa) \\
\Rightarrow & 1 + o(1) + \gamma(8\delta-6) - (\gamma(8\delta+4+6.5/(\delta-1)) + o(1)) \geq \Omega(\kappa) \\
\Rightarrow & 1 - \gamma(10 - 6.5(\delta-1)) \geq \Omega(\kappa) \\
\Rightarrow & \gamma \leq \frac{1-\Omega(\kappa)}{10 - 6.5(\delta-1)}
\end{aligned}
$$

Let $\gamma = \frac{1-O(\kappa)}{10+6.5/(\delta-1)}$, we have

$$
\begin{aligned}
\epsilon = & \frac{1}{8\delta-6} - \frac{1+o(1)}{(1+(8\delta-6)\gamma+o(1))(8\delta-6)} \\
= & \frac{1}{8\delta-6} - \frac{1+o(1)}{(1+(8\delta-6)\frac{1-O(\kappa)}{10+6.5/(\delta-1)}+o(1))(8\delta-6)} \\
= & \frac{1}{8\delta-6} - \frac{1}{(1+\frac{(8\delta-6)}{10+6.5/(\delta-1)})(8\delta-6)} - O(\kappa).
\end{aligned}
$$

When $2\delta = 4$, we get the maximized value $\epsilon = 2/53 - O(\kappa)$. □

5.3.4 Expansion for random $k\mathrm{CSP}$ instances

In this section, we prove Lemma 5.3.4, restated as follows.

Lemma 5.3.4 (restated). *Given β, η, K as in Theorem 5.3.2, with probability $1 - o(1)$, for all $2 \leq s \leq \eta n$, every set of s constraints involves more than $(K - \delta)s$ variables.*

Proof. Fix $2 \leq s \leq \eta n$, let us upperbound the probability that there is a set of s constraints containing at most $(K - \delta)s$ variables. Since there are $\binom{\beta n}{s}$ such sets, the probability is at most

$$\binom{\beta n}{s}\Pr[\text{the first } s \text{ constraints contain at most } (K-\delta)s \text{ variables}]$$
$$=\binom{\beta n}{s}\sum_{i=1}^{(K-\delta)s}\Pr[\text{the first } s \text{ constraints contain exactly } i \text{ variables}]$$

Fix a set T of i variables, let $p(s,i)$ be the number of s-tuples $(T_1, T_2, \cdots, T_s)$ where for each $1 \leq j \leq s$, T_j is a set of K variables, such that $\cup_{1\leq j\leq s} T_j = T$. We have

$$\Pr[\text{the first } s \text{ constraints contain exactly } i \text{ variables}] = \binom{n}{i}\cdot p(i,s)\Big/\binom{n}{K}^s.$$

To upperbound $p(i,s)$, we view the way to enumerating valid $(T_1, T_2, \cdots, T_s)$ as, to choose a multiset of Ks variables (each one from T) so that each element in T appears at least once in the multiset, then view each element in the multiset as a distinct element, and distribute these Ks elements to s sets, in a balanced way. Note that in this way, we are able to enumerate all the valid s-tuples (although some of them might be enumerated more than once). Since there are at most $\binom{Ks-1}{i-1} < \binom{Ks}{i}$ valid multisets, we have

$$p(i,s) \leq \binom{Ks}{i}(Ks)!/(K!)^s.$$

Therefore, we have

$$\binom{\beta n}{s}\Pr[\text{the first } s \text{ constraints contain at most } (K-\delta)s \text{ variables}]$$
$$=\binom{\beta n}{s}(Ks)!\cdot(K!)^{-s}\binom{n}{K}^{-s}\sum_{i=1}^{(K-\delta)s}\binom{n}{i}\binom{Ks}{i},$$

Note that when $K^2 s < \delta n$ and $i \leq (K-\delta)s$, we have $i < nKs/(n+Ks)$ (since $i \leq Ks(1-\delta/K) \leq Ks/(1+\delta/K) = nKs/(n+\delta n/K) < nKs/(n+Ks)$), and therefore

$$\frac{\binom{n}{i}\binom{Ks}{i}}{\binom{n}{i-1}\binom{Ks}{i-1}} = \frac{(n-i)(Ks-i)}{i^2} > 1 \qquad (\Leftarrow (n-i)(Ks-i) > i^2 \Leftarrow nKs > (n+Ks)i),$$

therefore the function $\binom{n}{i}\binom{Ks}{i}$ is increasing when $i \leq (K-\delta)s$, therefore

$$\binom{\beta n}{s}(Ks)!\cdot(K!)^{-s}\binom{n}{K}^{-s}\sum_{i=1}^{(K-\delta)s}\binom{n}{i}\binom{Ks}{i}$$

$$
\begin{aligned}
&\leq \binom{\beta n}{s}(Ks)! \cdot (K!)^{-s} \binom{n}{K}^{-s} \cdot Ks \cdot \binom{n}{(K-\delta)s} \binom{Ks}{(K-\delta)s} \\
&= \binom{\beta n}{s}(Ks)! \cdot (K!)^{-s} \binom{n}{K}^{-s} \cdot Ks \cdot \binom{n}{(K-\delta)s} \binom{Ks}{\delta s}.
\end{aligned}
$$

for $K \leq n^{1/2}$, we use the fact that $\binom{n}{K} \geq (n-K)^K/K! \geq n^K/3/((K/e)^K \cdot (5\sqrt{K})) = (en/K)^K/(15\sqrt{K})$ (since by Stirling's formula, we have $K! \leq 5\sqrt{K}(K/e)^K$), and again use the fact that $\sqrt{2\pi K}(K/e)^K \leq K! \leq 5\sqrt{K}(K/e)^K$, we bound the expression above by

$$
\begin{aligned}
&\left(\frac{e\beta n}{s}\right)^s \frac{5\sqrt{Ks}(Ks/e)^{Ks}}{(\sqrt{2\pi K}(K/e)^K)^s} \cdot \left(15\sqrt{K}\left(\frac{K}{en}\right)^K\right)^s \cdot Ks \cdot \left(\frac{en}{(K-\delta)s}\right)^{(K-\delta)s} \left(\frac{eK}{\delta}\right)^{\delta s} \\
\leq& 5(Ks)^{1.5} \cdot \left(\frac{15e\beta s^{\delta-1}K^{K+\delta}}{\sqrt{2\pi}n^{\delta-1}(K-\delta)^{K-\delta}\delta^\delta}\right)^s \\
\leq& 5(Ks)^{1.5} \cdot \left(\frac{15e^{1+\delta}\beta s^{\delta-1}K^{2\delta}}{\sqrt{2\pi}n^{\delta-1}\delta^\delta}\right)^s
\end{aligned}
$$

For $2 \leq s \leq \ln^2 n$, since $n^{\kappa-1} \leq 1/(10^8 \cdot (\beta K^{2\delta+0.75})^{1/(\delta-1)})$, we have $\beta^2 K^{4\delta+1.5}/n^{2(\delta-1)} \leq n^{-(2\delta-1)\kappa}$, we have

$$
\begin{aligned}
5(Ks)^{1.5} \cdot \left(\frac{15e^{1+\delta}\beta s^{\delta-1}K^{2\delta}}{\sqrt{2\pi}n^{\delta-1}\delta^\delta}\right)^s &\leq 5(Ks)^{1.5} \cdot \left(\frac{15e^{1+\delta}\beta s^{\delta-1}K^{2\delta}}{\sqrt{2\pi}n^{\delta-1}\delta^\delta}\right)^2 \\
&\leq \frac{5 \cdot 15e^{1+\delta}s^{\delta-1}}{\sqrt{2\pi}\delta^\delta n^{2(\delta-1)\kappa}} \leq O(n^{-(\delta-1)\kappa}).
\end{aligned}
$$

For $\ln^2 n < s \leq \eta n$, since $\eta \leq 1/(10^8 \cdot (\beta K^{2\delta+0.75})^{1/(\delta-1)})$, we get $\eta \leq 1/(10^8 \cdot (\beta K^{2\delta})^{1/(\delta-1)})$, and further we have $\beta K^{2\delta}\eta^{\delta-1} \leq \delta^\delta/(100 \cdot 15e^{1+\delta}/\sqrt{2\pi})$ for all $\delta > 5/4$. Therefore,

$$
\begin{aligned}
5(Ks)^{1.5} \cdot \left(\frac{15e^{1+\delta}\beta s^{\delta-1}K^{2\delta}}{\sqrt{2\pi}n^{\delta-1}\delta^\delta}\right)^s \leq 5(Ks)^{1.5}\left(\frac{s^{\delta-1}}{100(\eta n)^{\delta-1}}\right)^s &\leq 5 \cdot \left(\frac{s^{\delta-1}(Ks)^{(1.5/\ln^2 n)}}{100(\eta n)^{\delta-1}}\right)^s \\
&\leq 5 \cdot \left(\frac{2}{100}\right)^s \qquad \text{(by } s \leq \eta n \text{ and } Ks \leq n^2\text{)}
\end{aligned}
$$

Now, we upperbound probability that there exists a set of constraints of size $s \leq \eta n$

involving at most $(K-\delta)s$ variables by

$$
\begin{aligned}
&\sum_{s=2}^{\eta n} 5(Ks)^{1.5} \cdot \left(\frac{15e^{1+\delta}\beta s^{\delta-1}K^{2\delta+0.5}}{n^{\delta-1}\delta^{\delta}}\right)^s \\
=&\sum_{s=2}^{\ln^2 n} 5(Ks)^{1.5} \cdot \left(\frac{15e^{1+\delta}\beta s^{\delta-1}K^{2\delta+0.5}}{n^{\delta-1}\delta^{\delta}}\right)^s + \sum_{s=\ln^2 n+1}^{\eta n} 5(Ks)^{1.5} \cdot \left(\frac{15e^{1+\delta}\beta s^{\delta-1}K^{2\delta+0.5}}{n^{\delta-1}\delta^{\delta}}\right)^s \\
\leq&\sum_{s=2}^{\ln^2 n} O(n^{-\kappa/2}) + \sum_{s=\ln^2 n+1}^{\eta n} 5 \cdot (1/50)^s = o(1).
\end{aligned}
$$

□

Chapter 6

Lasserre integrality gaps for BALANCEDSEPARATOR and UNIFORMSPARSESTCUT

6.1 Introduction

Recall the problems BALANCEDSEPARATOR and UNIFORMSPARSESTCUT are defined previously by Definition 2.1.1 and Definition 2.1.2. For readers' convenience, we restate the definition as follows.

Definition 6.1.1. *Given an undirected graph $G = (V, E)$ and $0 < \tau < 0.5$, the goal of the τ vs $1 - \tau$ BALANCEDSEPARATOR problem is to find a set $A \subseteq V$ such that $\tau|V| \le |A| \le (1-\tau)|V|$, while $\mathrm{edges}(A, V \setminus A)$ is minimized. Here $\mathrm{edges}(A, B)$ is the number of edges in E that cross the cut (A, B).*

The goal of the UNIFORMSPARSESTCUT problem is to find a set $\emptyset \subsetneq A \subsetneq V$ such that the sparsity

$$\frac{\mathrm{edges}(A, V \setminus A)}{|A||V \setminus A|}$$

is minimized.

Despite extensive research, there are still huge gaps between the known approximation algorithms and inapproximability results for these problems. The best algorithms, based on semidefinite relaxations (SDPs) with triangle inequalities, give a $O(\sqrt{\log n})$ approximation to both problems [21]. On the inapproximability side, a Polynomial Time Approximation Scheme (PTAS) is ruled out for both problems assuming 3-SAT does not have randomized subexponential-time algorithms [11]. In this chapter, our focus is on the UNIFORMSPARSESTCUT problem; the general SPARSESTCUT problem has been shown to not admit a constant-factor approximation algorithm under the Unique Games Conjecture [64, 144, 136].

It is known that the SDP used in [21] cannot give a constant factor approximation for UNIFORMSPARSESTCUT [79]. Integrality gaps are also known for stronger SDPs: super-constant factor integrality gaps for both BALANCEDSEPARATOR and UNIFORMSPARSESTCUT are known for the Sherali-Adams+SDP hierarchy for a super-constant number of rounds [192].

However, if we turn to the stronger Lasserre SDP hierarchy, for both BALANCEDSEPARATOR and UNIFORMSPARSESTCUT, integrality gaps were not known even for a small constant number of rounds (before this thesis). It was not (unconditionally) ruled out, for example, that $1/\epsilon^{O(1)}$ rounds of the hierarchy could give a $(1+\epsilon)$-approximation algorithm, thereby giving a PTAS. On the algorithmic side, [110] recently showed that for these problems, SDPs using $O(r/\epsilon^2)$ rounds of the Lasserre hierarchy have an integrality gap at most $(1+\epsilon)/\min\{1, \lambda_r\}$. Here λ_r is the r-th smallest eigenvalue of the normalized Laplacian of the graph. This result implies an approximation scheme for these problems with runtime parameterized by the graph spectrum.

6.1.1 Our contributions

In this chapter, we study integrality gaps for the Lasserre SDP relaxations for BALANCEDSEPARATOR and UNIFORMSPARSESTCUT. As mentioned before, APX-hardness is not known for these two problems, even assuming the Unique Games Conjecture. (Superconstant hardness results are known based on a strong intractability assumption concerning the Small Set Expansion problem [194].) In contrast, we show that linear-round Lasserre SDP has an integrality gap bounded away from 1, and thus fails to give a factor α-approximation for some absolute constant $\alpha > 1$. Specifically, we prove the following two theorems.

Theorem 6.1.2 (Informal version of Theorem 6.3.1)**.** *For $0.45 < \tau < 0.5$, there are linear-round Lasserre gap instances for the τ vs $(1-\tau)$ BALANCEDSEPARATOR problem, such that the integral optimal solution is at least $(1+\epsilon(\tau))$ times the SDP solution, where $\epsilon(\tau) > 0$ is a constant dependent on τ.*

Theorem 6.1.3 (Informal version of Theorem 6.4.1)**.** *There are linear-round Lasserre gap instances for the* UNIFORMSPARSESTCUT *problem, such that the integral optimal solution is at least* $(1+\epsilon)$ *times the SDP solution, for some constant* $\epsilon > 0$*.*

6.1.2 Our techniques

All of our gap results are based on Schoenebeck's ingenious Lasserre integrality gap for 3-XOR [204]. For BALANCEDSEPARATOR and UNIFORMSPARSESTCUT, we use the ideas in [11] to build gadget reductions and combine them with Schoenebeck's gap instance. [11] designed gadget reductions from Khot's quasi-random PCP [137] in order to show APX-Hardness of the two problems. If we view the Lasserre hierarchy as a computational model (as suggested in [220]), we can view Schoenebeck's construction as playing the role of a quasi-random PCP in the Lasserre model. Our gadget reductions, therefore, bear some resemblance to the ones in [11], though the analysis is different due to different random structures of the PCPs. We feel our reductions are slightly simpler than the ones in [11], although we need some additional tricks to make the reductions have only linear blowup. This latter feature is needed in order to get Lasserre SDP gaps for a linear number of rounds. We are also able to make the gap instance graphs have only **constant degree**, while the reductions in [11] give graphs with unbounded degree.

Also, unlike 3-XOR, for balanced separator there is a global linear constraint (stipulating the balance of the cut), and our Lasserre solution must also satisfy a lifted form of this constraint [158]. We make a general observation that such constraints can be easily lifted to the Lasserre hierarchy when the vectors in our construction satisfy a related linear constraint. This observation applies to constraints given by any polynomials, and to our knowledge, was not made before. It simplifies the task of constructing legal Lasserre vectors in such cases.

6.2 Preliminaries on Lasserre SDPs for BALANCEDSEPARATOR and UNIFORMSPARSESTCUT

In this section, we begin with a general description of semidefinite programming relaxations from the Lasserre hierarchy, followed by a useful observation about constructing feasible solutions for such a SDP. We then discuss the specific SDP relaxations for our problems of interest. Finally, we recall Schoenebeck's Lasserre integrality gaps [204] in a form convenient for our later use.

6.2.1 Useful theorems about the Lasserre hierarchy

Consider a binary programming problem with polynomial objective function P and a single constraint expressed as a polynomial Q:

$$\begin{array}{rl} \text{Minimize/Maximize} & \sum_{T\in\binom{[n]}{\le d}} P(T)\prod_{j\in T} x_j \\ \text{subject to} & \sum_{T\in\binom{[n]}{\le d}} Q(T)\prod_{j\in T} x_j \ge 0, \\ & x_i\in\{0,1\} \qquad \text{for all } i\in[n]. \end{array} \tag{6.1}$$

It is easy to see that this captures all problems we consider in this chapter: BALANCED-SEPARATOR (Section 6.2.2.1) and UNIFORMSPARSESTCUT (Section 6.2.2.2).

We now define the Lasserre hierarchy semidefinite program relaxation for the above integer program. It is easily seen that the below is a relaxation by taking $\overline{\boldsymbol{U}}_A = x_A\mathbf{I}$ and $Y_A = \sqrt{Q(x)}\,\overline{\boldsymbol{U}}_A$ where $x\in\{0,1\}^n$ is a feasible solution to (6.1), $x_A = \prod_{i\in A} x_i$, and $\mathbf{I}$ is any fixed unit vector.

Proposition 6.2.1. *For any positive integer $r \ge d$, r rounds of Lasserre Hierarchy relaxation [158] of* (6.1) *is given by the following semidefinite programming formulation:*

$$\begin{array}{rll} \textit{Minimize/Maximize} & \sum_T P(T)\left\|\overline{\boldsymbol{U}}_T\right\|^2 & \\ \textit{subject to} & \|\overline{\boldsymbol{U}}_\emptyset\|^2 = 1, & \\ & \langle\overline{\boldsymbol{U}}_A,\overline{\boldsymbol{U}}_B\rangle = \|\overline{\boldsymbol{U}}_{A\cup B}\|^2 & \textit{for all } A,B \textit{ with } |A\cup B|\le 2r, \\ & \sum_{S\in\binom{[n]}{\le d}} Q(S)\langle\overline{\boldsymbol{U}}_S,\overline{\boldsymbol{U}}_{A\cup B}\rangle = \langle Y_A, Y_B\rangle, & \\ & \langle Y_A, Y_B\rangle = \|Y_{A\cup B}\|^2 & \textit{for all } A,B \textit{ with } |A\cup B|\le 2(r-d). \\ & \overline{\boldsymbol{U}}_A, Y_B \in \mathbb{R}^\Upsilon. & \end{array} \tag{6.2}$$

Proof. Given $y\in\mathbb{R}^{\binom{[n]}{\le 2r}}$, let $\mathbb{M}(y)\in\mathsf{Sym}(\binom{[n]}{\le r})$ be the moment matrix whose rows and columns correspond to subsets of size $\le r$. The entry at row S and column T of $\mathbb{M}(y)$ is given by $y_{S\cup T}$. For any multilinear polynomial P of degree-d, let $P * y\in\mathbb{R}^{\binom{[n]}{\le 2r-d}}$ be the vector whose entry corresponding to subset S is given by $\sum_T P_T y_{S\cup T}$. The Lasserre Hierarchy relaxation [158] of (6.1) is given by:

$$\begin{array}{rl} \text{Minimize/Maximize} & \sum_T P(T) y_T \\ \text{subject to} & y_\emptyset = 1, \\ & \mathbb{M}(y)\succeq 0, \\ & \mathbb{M}(Q*y)\succeq 0. \end{array} \tag{6.3}$$

Proof of (6.2) $\implies$ (6.3). Given feasible solution for (6.2), let $y_S \triangleq \|\overline{\boldsymbol{U}}_S\|^2$ and $z_S \triangleq \|Y_S\|^2$. We have $y_\emptyset = 1$ and $\sum_T P(T)y_T = \sum_T P(T)\|\overline{\boldsymbol{U}}_S\|^2$. Observe that $y_{S\cup T} = \|\overline{\boldsymbol{U}}_{S\cup T}\|^2 = \langle \overline{\boldsymbol{U}}_S, \overline{\boldsymbol{U}}_T\rangle$ therefore $\mathbb{M}(y) \succeq 0$. With a similar reasoning, we also have $\mathbb{M}(z) \succeq 0$. Finally, for any S:

$$(Q * y)_S = \sum_T Q(T) y_{S\cup T} = \sum_T Q(T)\langle \overline{\boldsymbol{U}}_T, \overline{\boldsymbol{U}}_S\rangle = \|Y_S\|^2 = z_S,$$

which implies $z = Q * y$. Hence y is a feasible solution for (6.3).

Proof of (6.3) $\implies$ (6.2). Let y be a feasible solution for (6.3). Define $z \triangleq Q * y$. Since $\mathbb{M}(y) \succeq 0$ (resp. $\mathbb{M}(z) \succeq 0$), there exists a matrix $\overline{\boldsymbol{U}} = [\overline{\boldsymbol{U}}_S]_S$ (resp. $Y = [Y_S]_S$) such that $\mathbb{M}(y) = \overline{\boldsymbol{U}}^T\overline{\boldsymbol{U}}$ (resp. $\mathbb{M}(z) = Y^T Y$). It is easy to see that $\langle \overline{\boldsymbol{U}}_S, \overline{\boldsymbol{U}}_T\rangle = y_{S\cup T}$ and $\langle Y_S, Y_T\rangle = z_{S\cup T}$. Therefore:

- $\sum_T P(T)\|\overline{\boldsymbol{U}}_T\|^2 = \sum_T P(T) y_T$.
- $\|\overline{\boldsymbol{U}}_\emptyset\|^2 = y_\emptyset = 1$.
- $\langle \overline{\boldsymbol{U}}_S, \overline{\boldsymbol{U}}_T\rangle = y_{S\cup T} = \|\overline{\boldsymbol{U}}_{S\cup T}\|^2$ (similar for Y).
- For any S, $\sum_T Q(T)\langle \overline{\boldsymbol{U}}_T, \overline{\boldsymbol{U}}_S\rangle = \sum_T Q(T) y_{S\cup T} = (Q * y)_S = z_S = \|Y_S\|^2$.

Therefore $(\overline{\boldsymbol{U}}, Y)$ is a feasible solution for (6.2) with same objective value, completing our proof. $\square$

Note that a straightforward verification of last two constraints requires the construction of vectors Y_A in addition to $\overline{\boldsymbol{U}}_A$. Below we give an easier way to verify these last two constraints without having to construct Y_A's. This greatly simplifies our task of constructing Lasserre vectors for the lifting of global balance constraints.

Theorem 6.2.2. *Given vectors $\overline{\boldsymbol{U}}_T$ for all $T \in \binom{[n]}{\le 2r}$ satisfying the first two constraints of* (6.2)*, if there exists a non-negative real $\delta > 0$ such that*

$$\sum_{S\in\binom{[n]}{\le d}} Q(S)\overline{\boldsymbol{U}}_S = \delta \cdot \overline{\boldsymbol{U}}_\emptyset \tag{6.4}$$

then these vectors form (part of) a feasible solution to (6.2).

Proof. Consider the following vectors. For each A with $|A| \leq r$, let $Y_A = \sqrt{\delta} \cdot \overline{\boldsymbol{U}}_A$. By construction, these vectors satisfy the $\langle Y_A, Y_B \rangle = \|Y_{A \cup B}\|^2$ constraints since $\langle \overline{\boldsymbol{U}}_A, \overline{\boldsymbol{U}}_B \rangle = \|\overline{\boldsymbol{U}}_{A \cup B}\|^2$. Now we verify the other constraint:

$$\sum_{S \in \binom{[n]}{\leq d}} Q(S) \langle \overline{\boldsymbol{U}}_S, \overline{\boldsymbol{U}}_{A \cup B} \rangle = \left\langle \sum_{S \in \binom{[n]}{\leq d}} Q(S) \overline{\boldsymbol{U}}_S, \overline{\boldsymbol{U}}_{A \cup B} \right\rangle$$
$$= \langle \delta \overline{\boldsymbol{U}}_{\emptyset}, \overline{\boldsymbol{U}}_{A \cup B} \rangle = \delta \langle \overline{\boldsymbol{U}}_A, \overline{\boldsymbol{U}}_B \rangle = \langle Y_A, Y_B \rangle.$$

□

6.2.2 Lasserre SDP for graph partitioning problems

In light of Theorem 6.2.2, to show good solutions for the Lasserre SDP for our problems of interest, we only need to show good solutions for the following SDPs.

6.2.2.1 BALANCEDSEPARATOR

The standard integer programming formulation of BALANCEDSEPARATOR is shown in the left part of Figure 6.1. The r round SDP relaxation Ψ_1 (shown in the right part of Figure 6.1) has a vector $\overline{\boldsymbol{U}}_S$ for each subset $S \subseteq V$ with $|S| \leq r$. In an integral solution, the intended value of $\overline{\boldsymbol{U}}_{\{u\}}$ is $x_u \overline{\boldsymbol{U}}_{\emptyset}$ for some fixed unit vector $\overline{\boldsymbol{U}}_{\emptyset}$, and that of $\overline{\boldsymbol{U}}_S$ is $\left(\prod_{u \in S} x_u\right) \overline{\boldsymbol{U}}_{\emptyset}$.

6.2.2.2 UNIFORMSPARSESTCUT

The UNIFORMSPARSESTCUT problem asks to minimize the value of the quadratic integer program shown in the left part of Figure 6.2 over all $\tau \in \{1/n, 2/n, \ldots, \lfloor n/2 \rfloor\}$. The corresponding SDP relaxation Ψ_2 is to minimize the value of the SDP shown in the right part of Figure 6.2 over all $\tau \in \{1/n, 2/n, \ldots, \lfloor n/2 \rfloor\}$.

Remark Prior to this thesis, known lower bounds [79, 130] on the integrality gap of UNIFORMSPARSESTCUT problem used a **weaker** relaxation, where the last two equality constraints in Ψ_2 of Figure 6.2 are replaced by the following instead:

$$\sum_{u<v} \|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v\}}\|^2 = 1$$

IP	SDP Relaxation Ψ_1
$$\begin{aligned} \text{Minimize} \quad & \sum_{(u,v)\in E} (x_u - x_v)^2 \\ \text{s.t.} \quad & \tau\lvert V\rvert \le \sum_{u\in V} x_u \le (1-\tau)\lvert V\rvert \\ & x_u \in \{0,1\} \quad \forall u \in V \end{aligned}$$	$$\begin{aligned} \text{Minimize} \quad & \sum_{(u,v)\in E} \left\lVert \overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v\}} \right\rVert^2 \\ \text{s.t.} \quad & \langle \overline{\boldsymbol{U}}_{S_1}, \overline{\boldsymbol{U}}_{S_2} \rangle \ge 0 \text{ for all } S_1, S_2 \\ & \langle \overline{\boldsymbol{U}}_{S_1}, \overline{\boldsymbol{U}}_{S_2} \rangle = \langle \overline{\boldsymbol{U}}_{S_3}, \overline{\boldsymbol{U}}_{S_4} \rangle \\ & \qquad \text{for all } S_1 \cup S_2 = S_3 \cup S_4 \\ & \left\lVert \overline{\boldsymbol{U}}_{\emptyset} \right\rVert^2 = 1 \\ & \sum_v \overline{\boldsymbol{U}}_{\{v\}} = \tau' \lvert V\rvert \overline{\boldsymbol{U}}_{\emptyset} \\ & \qquad \text{for some } \tau \le \tau' \le 1-\tau \end{aligned}$$

Figure 6.1: IP and SDP relaxations for BALANCEDSEPARATOR.

We can solve Ψ_1 by first enumerating over all $\tau' \in \{1/n, 2/n, \ldots, 1\} \cap [\tau, 1-\tau]$ and then choosing τ which minimizes the objective function. Note that the resulting relaxation is **stronger** than usual Lasserre Hierarchy relaxation.

with the objective function being simply $\sum_{(u,v)\in E} \|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v\}}\|^2$.

6.2.3 Lasserre Gaps for 3-XOR from Schoenebeck

We start by defining the 3-XOR problem.

Definition 6.2.3. *An instance Φ of* 3-XOR *is a set of constraints $C_1, C_2, \cdots, C_m$ where each constraint C_i is over* 3 *distinct variables x_{i_1}, x_{i_2}, and x_{i_3}, and is of the form $x_{i_1} \oplus x_{i_2} \oplus x_{i_3} = b_i$ for some $b_i \in \{0,1\}$.*

A random instance of 3-XOR *is sampled by choosing each constraint C_i uniform independently from the set of possible constraints.*

We will make use of the following fundamental result of Schoenebeck.

Theorem 6.2.4 ([204])**.** *For every large enough constant $\beta > 1$, there exists $\eta > 0$, such that with probability $1 - o(1)$, a random* 3-XOR *instance Φ over $m = \beta n$ constraints and n variables cannot be refuted by the SDP relaxation obtained by ηn rounds of the Lasserre hierarchy, i.e. there are vectors $\boldsymbol{W}_{(S,\alpha)}$ for all $|S| \le \eta n$ and all $\alpha : S \to \{0,1\}$, such that*

IP	SDP Relaxation Ψ_2
Minimize $\dfrac{1}{\|V\|^2\tau(1-\tau)} \sum_{(u,v)\in E} (x_u - x_v)^2$ s.t. $\sum_u x_u = \tau\|V\|$ $x_u \in \{0,1\} \quad \forall u \in V$	Minimize $\sum_{(u,v)\in E} \dfrac{1}{\|V\|^2\tau(1-\tau)} \left\\|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v\}}\right\\|^2$ s.t. $\langle \overline{\boldsymbol{U}}_{S_1}, \overline{\boldsymbol{U}}_{S_2} \rangle \geq 0$ for all S_1, S_2 $\langle \overline{\boldsymbol{U}}_{S_1}, \overline{\boldsymbol{U}}_{S_2} \rangle = \langle \overline{\boldsymbol{U}}_{S_3}, \overline{\boldsymbol{U}}_{S_4} \rangle$ for all $S_1 \cup S_2 = S_3 \cup S_4$ $\left\\|\overline{\boldsymbol{U}}_{\emptyset}\right\\|^2 = 1$ $\sum_v \overline{\boldsymbol{U}}_{\{v\}} = \tau\|V\|\overline{\boldsymbol{U}}_{\emptyset}$

Figure 6.2: IP and SDP relaxations for UNIFORMSPARSESTCUT.

We can solve Ψ_2 by first enumerating over all $\tau \in \{1/n, 2/n, \ldots, (n-1)/n\}$ and then choosing τ which minimizes the objective function. Note that the resulting relaxation is **stronger** than usual Lasserre Hierarchy relaxation.

(i) *the value of the solution is perfect:* $\sum_{i=1}^{m} \sum_{\alpha:\alpha(x_{i_1})\oplus\alpha(x_{i_2})\oplus\alpha(x_{i_3})=b_i} \left\|\boldsymbol{W}_{(\{x_{i_1},x_{i_2},x_{i_3}\},\alpha)}\right\|^2 = m$*;*

(ii) $\langle \boldsymbol{W}_{(S_1,\alpha_1)}, \boldsymbol{W}_{(S_2,\alpha_2)} \rangle \geq 0$ *for all* $S_1, S_2, \alpha_1, \alpha_2$*;*

(iii) $\langle \boldsymbol{W}_{(S_1,\alpha_1)}, \boldsymbol{W}_{(S_2,\alpha_2)} \rangle = 0$ *if* $\alpha_1(S_1 \cap S_2) \neq \alpha_2(S_1 \cap S_2)$*;*

(iv) $\langle \boldsymbol{W}_{(S_1,\alpha_1)}, \boldsymbol{W}_{(S_2,\alpha_2)} \rangle = \langle \boldsymbol{W}_{(S_3,\alpha_3)}, \boldsymbol{W}_{(S_4,\alpha_4)} \rangle$ *for all* $S_1 \cup S_2 = S_3 \cup S_4$ *and* $\alpha_1 \circ \alpha_2 = \alpha_3 \circ \alpha_4$*. Here, when* $\alpha_1(S_1 \cap S_2) = \alpha_2(S_1 \cap S_2)$*,* $\alpha_1 \circ \alpha_2$ *is naturally defined as the mapping from* $S_1 \cap S_2$ *to* $\{0,1\}$ *such that its restriction to* S_1 *equals* α_1 *and its restriction to* S_2 *equals* α_2*. We make similar definition for* $\alpha_3 \circ \alpha_4$*.*

(v) $\sum_{\alpha:S\to\{0,1\}} \left\|\boldsymbol{W}_{(S,\alpha)}\right\|^2 = 1$ *for all* S*.*

Note that indeed we have for every S, $\sum_{\alpha:S\to\{0,1\}} \boldsymbol{W}_{(S,\alpha)} = \boldsymbol{W}_{(\emptyset,\emptyset)}$. This is because $\left\|\boldsymbol{W}_{(\emptyset,\emptyset)}\right\|^2 = 1$ and

$$\left\langle \Big(\sum_{\alpha:S\to\{0,1\}} \boldsymbol{W}_{(S,\alpha)} \Big), \boldsymbol{W}_{(\emptyset,\emptyset)} \right\rangle = \sum_{\alpha:S\to\{0,1\}} \langle \boldsymbol{W}_{(S,\alpha)}, \boldsymbol{W}_{(\emptyset,\emptyset)} \rangle = \sum_{\alpha:S\to\{0,1\}} \left\|\boldsymbol{W}_{(S,\alpha)}\right\|^2 = 1.$$

Observation 6.2.5. *In the construction of Theorem 6.2.4, the vectors* $\boldsymbol{W}$ *satisfy the following property. For any constraint* C_i *over set of variables* S_i*, the vectors corresponding* *to all satisfying partial assignments of S_i sums up to $\boldsymbol{W}_\emptyset$:*

$$\sum_{\alpha:S_i\to\{0,1\}\wedge C_i(\alpha)=1} \boldsymbol{W}_{(S_i,\alpha)} = \boldsymbol{W}_\emptyset.$$

6.3 Gaps for BALANCEDSEPARATOR

In this section, we prove Theorem 6.1.2. We state the theorem in detail as follows.

Theorem 6.3.1. *For large enough constants β, M, for all $0.45 < \tau < 0.5$, and for infinitely many positive integer N's, there is an N-vertex instance $\mathcal{H}_\Phi$ for the τ vs. $(1-\tau)$ BALANCEDSEPARATOR problem, such that the optimal solution is at least $4(3\tau-\tau^3)/5 - O(1/\sqrt{\beta} + 1/M)$ times the best solution of the $\Omega(N)$-round Lasserre SDP relaxation. Moreover, the solution for Lasserre SDP relaxation is a fractional $(0.5 - O(1/M))$ vs. $(0.5 + O(1/M))$ balanced separator.*

The rest of this section is dedicated to the proof of Theorem 6.3.1. In Section 6.3.1, we will describe how to get a BALANCEDSEPARATOR instance from a 3-XOR instance. Then, we will show that when the 3-XOR instance is random, the corresponding BALANCEDSEPARATOR instance is a desired gap instance. This is done by showing there is an SDP solution with good objective value (completeness part, Lemma 6.3.2 in Section 6.3.2) while the instance in fact has not great integral solution (soundness part, Lemma 6.3.4 in Section 6.3.3). The completeness part relies Theorem 6.2.4 – we use the 3-XOR vectors (which exist for random instances by the theorem) to construct BALANCEDSEPARATOR vectors. In the soundness part, we first prove two pseudorandom structural properties exhibited in the random 3-XOR instances (Lemma 6.3.3), and then prove that any 3-XOR with these two properties leads to a BALANCEDSEPARATOR instance with bad integral optimum by our construction. Finally, in Section 6.3.4, we slightly twist our gap instance in order to make its vertex degree bounded.

6.3.1 Reduction

Given a 3-XOR instance Φ with $m = \beta n$ constraints and n variables, we build a graph $\mathcal{H}_\Phi = (\mathcal{V}_\Phi, \mathcal{E}_\Phi)$ for BALANCEDSEPARATOR as follows.

$\mathcal{H}_\Phi$ consists of an almost bipartite graph $H_\Phi = (L_\Phi, R_\Phi, E_\Phi)$ (obtained by replacing each right vertex of a bipartite graph by a clique), an expander Z_r, and edges between L_Φ and Z_r.

The left side L_Φ of H_Φ contains $4m = 4\beta n$ vertices, each corresponds to a pair of a constraint and a satisfying partial assignment for the constraint, i.e.

$$L_\Phi = \{(C_i, \alpha) | \alpha : \{x_{i_1}, x_{i_2}, x_{i_3}\} \to \{0, 1\}, C_i(\alpha) = 1\}.$$

The right side R_Φ of H_Φ contains $2n$ cliques, each is of size $M\beta$, and corresponds to one of the $2n$ literals, i.e.

$$R_\Phi = \cup_{j, \alpha : \{x_j\} \to \{0,1\}} C_{(x_j, \alpha)},$$

where

$$C_{(x_j,\alpha)} = \{(x_j, \alpha, t) | 1 \le t \le M\beta\}.$$

Call $(x_j, \alpha, 1)$ the *representative vertex* of $C_{(x_j,\alpha)}$. Besides the clique edges, we connect a left vertex (C_i, α) and a right representative vertex $(x_j, \alpha', 1)$ if x_j is accessed by C_i and α' is consistent with α, i.e.

$$E_\Phi = \{\text{clique edges}\} \cup \{\{(C_i, \alpha), (x_j, \alpha', 1)\} | x_j \in \{x_{i_1}, x_{i_2}, x_{i_3}\}, \alpha(x_j) = \alpha'(x_j)\}.$$

Now we have finished the definition of H_Φ. To get $\mathcal{H}_\Phi$, we add an $O(M)$-regular expander Z_r of size $m = \beta n$ and edge expansion M. (I.e. the degree of each vertex in Z_r is $O(M)$, and each subset $T \subseteq Z_r$ ($|T| \le |Z_r|/2$) has at least $|T| \cdot M$ edges connecting to $Z_r \setminus T$. For more discuss on the definitions and applications of expander graphs, please refer to, e.g. , [121].) We connect each vertex in L_Φ to two different vertices in Z_r, so that each vertex in Z_r has the same number of neighbors in L_Φ (this number should be $4\beta n \cdot 2/(\beta n) = 8$). In other words, if we view each vertex in L_Φ as an undirected edge between its two neighbors in Z_r, the graph should be a regular graph.

The whole construction is shown in Figure 6.3. Our construction is very similar to the one in [11], but there are some technical differences. Instead of having cliques in R_Φ, [11] has clusters of vertices with no edges connecting them. Also, in our construction, the vertices in L_Φ are connected to the representative vertices in R_Φ only, while in [11], all the vertices in the right clusters could be connected to the left side. The most important difference is that in our way, the cliques are of constant size, while the clusters in [11] has superconstantly many vertices. This means that our reduction blows up the instance size only by a constant factor, therefore we are able to get linear round Lasserre gap.

Observe that there are $|L_\Phi| + |R_\Phi| + |Z_r| = 4m + 2Mm + m = (2M + 5)m$ vertices in $\mathcal{H}_\Phi$.

In the following two subsections, we will prove the completeness lemma (Lemma 6.3.2, which states that there is an SDP solution with a good objective value) and the soundness lemma (Lemma 6.3.4, which states that every integral solution has a bad objective value). Combining the two lemmas, we prove our main integrality gap theorem for BALANCED-SEPARATOR as follows.

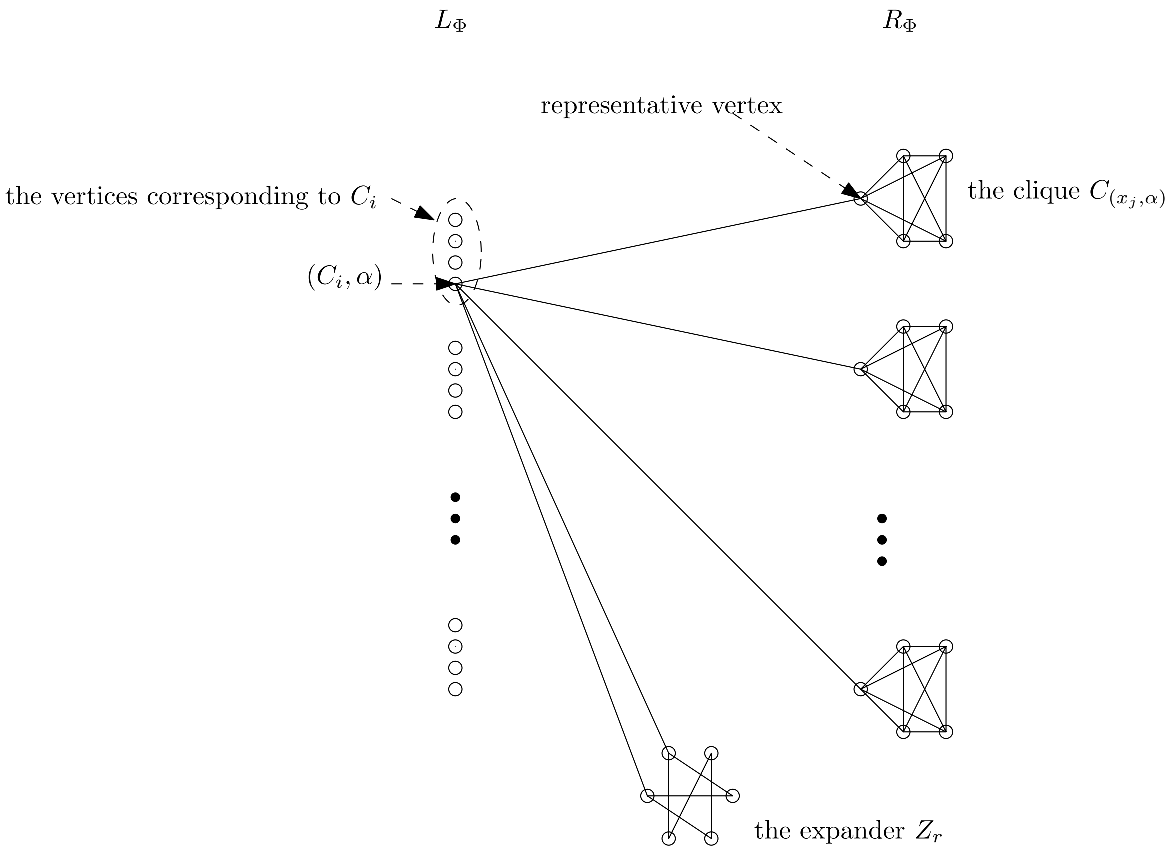

Figure 6.3: The reduction for BALANCEDSEPARATOR.

Note that the incident edges are drawn for only one of the vertices in L_Φ, while others can be drawn similarly.

of Theorem 6.3.1 from Lemma 6.3.2 and Lemma 6.3.4. Let β, M be large enough constants. Let Φ be a random 3-XOR instance over $m = \beta n$ constraints and n variables.

By Theorem 6.2.4 we know that, with probability $1 - o(1)$, Φ admits a perfect solution for $\Omega(n)$-round Lasserre SDP relaxation. Therefore, by Lemma 6.3.2, with probability $1 - o(1)$, $\Omega(n)$-round SDP relaxation Ψ_1 with parameter $\tau = 0.5 - O(1/M)$ for the BALANCEDSEPARATOR instance $\mathcal{H}_\Phi$ has a solution of value $5m$. On the other hand, by Lemma 6.3.4, with probability $1-o(1)$, for $\tau > 1/3$, every τ vs. $(1-\tau)$ balanced separator has at least $4m(3\tau - \tau^3 - O(1/\sqrt{\beta}) - O(1/M))$ edges in the cut.

Therefore, with probability $1 - o(1)$, when $\tau > 1/3$, the ratio between the optimal integral solution (to $\mathcal{H}_\Phi$) and the optimal $\Omega(n)$-round Ψ_1 solution is at least $4(3\tau - \tau^3)/5 -$

$O(1/\sqrt{\beta} + 1/M)$. This ratio is greater than 1.007 when $\tau > 0.45$ and β and M are large enough. By our observation in Section 6.2.2.1, this gap also holds for the Lasserre SDP relaxation. $\square$

Let Δ be the maximum number of occurrences of any variable in Φ. By our construction, the graph has degree $\Theta(M + \Delta)$. When $\beta = O(1)$, we have $\Delta = \Theta(\log n / \log \log n)$ with probability $1 - o(1)$ (see, e.g. [96]). This means that our graph does not have the desired constant-degree property. However, since there are few edges incident to vertices with superconstant degree, we can simply remove all these edges to get a constant-degree graph, while the completeness and soundness are still preserved. We will discuss this in more details in Section 6.3.4.

6.3.2 Completeness : good SDP solution

Lemma 6.3.2 (Completeness)**.** *If the* 3-XOR *instance* Φ *admits perfect solution for* r*-round Lasserre SDP relaxation, then the* $r/3$*-round SDP relaxation* Ψ_1 *(in Figure 6.1) with parameter* $\tau = 0.5 - O(1/M)$ *for the* BALANCEDSEPARATOR *instance* $\mathcal{H}_\Phi$ *has a solution of value* $5m$*.*

Proof. We define a set of vectors (i.e. a solution to Ψ_1) using the vectors given in Theorem 6.2.4, as follows.

For each set $S \subseteq L_\Phi \cup R_\Phi \cup Z_r$ with $|S| \leq r/3$, we define the vector $\overline{\boldsymbol{U}}_S$ as follows. If $S \cap Z_r \neq \emptyset$, let $\overline{\boldsymbol{U}}_S = \mathbf{0}$. If $S \cap Z_r = \emptyset$, suppose that $S \cap L_\Phi$ contains

$$(C_{i_1}, \alpha_1), (C_{i_2}, \alpha_2), \cdots, (C_{i_{r_1}}, \alpha_{r_1}),$$

$S \cap R_\Phi$ contains

$$(x_{j_1}, \alpha'_1, t_1), (x_{j_2}, \alpha'_2, t_2), \cdots, (x_{j_{r_2}}, \alpha'_{r_2}, t_{r_2}),$$

we have $r_1 + r_2 = |S|$. Let S' be the set of variables accessed by $C_{i_1}, \cdots C_{i_2}$ together with $x_{j_1}, \cdots, x_{j_{r_2}}$. Note that $|S'| \leq 3r_1 + r_2 \leq 3|S| \leq r$. If there is no contradiction among the partial assignments α_i's and α'_i's (i.e. there are not two of them assigning the same variable to different values), we can define

$$\alpha = \alpha_1 \circ \alpha_2 \circ \cdots \alpha_{r_1} \circ \alpha'_1 \circ \alpha'_2 \circ \cdots \circ \alpha'_{r_2}.$$

and let $\overline{\boldsymbol{U}}_S = \boldsymbol{W}_{(S',\alpha)}$, otherwise we let $\overline{\boldsymbol{U}}_S = \mathbf{0}$.

We first check that the first 3 constraints in relaxation Ψ_1 are satisfied.

- For two sets S_1, S_2, either at least one of the vectors $\overline{\boldsymbol{U}}_{S_1}, \overline{\boldsymbol{U}}_{S_2}$ is $\mathbf{0}$ (therefore their inner-product is 0), or $\overline{\boldsymbol{U}}_{S_1} = \boldsymbol{W}_{S_1',\alpha_1}, \overline{\boldsymbol{U}}_{S_2} = \boldsymbol{W}_{S_2',\alpha_2}$ for some $S_1', S_2', \alpha_1, \alpha_2$ and $\langle \overline{\boldsymbol{U}}_{S_1}, \overline{\boldsymbol{U}}_{S_2} \rangle = \langle \boldsymbol{W}_{S_1',\alpha_1}, \boldsymbol{W}_{S_2',\alpha_2} \rangle \geq 0$.
- For any S_1, S_2, S_3, S_4 such that $S_1 \cup S_2 = S_3 \cup S_4$, either the set of partial assignments in $S_1 \cup S_2 = S_3 \cup S_4$ are consistent with each other, in which case we have $\overline{\boldsymbol{U}}_{S_1 \cup S_2} = \overline{\boldsymbol{U}}_{S_3 \cup S_4} = \boldsymbol{W}_{S,\alpha}$ where S is the union of all the variables included in $S_1 \cup S_2$ and α is the concatenation of the partial assignments in $S_1 \cup S_2$; or we have $\overline{\boldsymbol{U}}_{S_1 \cup S_2} = \overline{\boldsymbol{U}}_{S_3 \cup S_4} = \mathbf{0}$.
- $\|\boldsymbol{U}_{\emptyset}\|^2 = \left\|\boldsymbol{W}_{(\emptyset,\emptyset)}\right\|^2 = 1$.

Now we check that the balance condition (the last constraint in relaxation Ψ_1) is satisfied. We will prove that

$$\sum_v \overline{\boldsymbol{U}}_{\{v\}} = (M+1)m\overline{\boldsymbol{U}}_{\emptyset}.$$

Since there are $(2M+5)m$ vertices in $\mathcal{H}_\Phi$, this shows that the solution is feasible for Φ_1 with $\tau = 0.5 - O(1/M)$. Using 6.2.5, we see that $\sum_{(C_i,\alpha)\in L_\Phi} \overline{\boldsymbol{U}}_{\{(C_i,\alpha)\}} = \sum_{C_i} \overline{\boldsymbol{U}}_{\emptyset} = m\overline{\boldsymbol{U}}_{\emptyset}$. Similarly

$$\begin{aligned}\sum_{(x_j,\alpha,t)\in R_\Phi} \overline{\boldsymbol{U}}_{\{(x_j,\alpha,t)\}} =& \sum_{j=1}^{n} \sum_{\alpha:\{x_j\}\to\{0,1\}} \sum_{t=1}^{\beta M} \overline{\boldsymbol{U}}_{\{(x_j,\alpha,t)\}} = \beta M \cdot \sum_{j=1}^{n} \sum_{\alpha:\{x_j\}\to\{0,1\}} \overline{\boldsymbol{U}}_{\{(x_j,\alpha,1)\}} \\ =& \beta M n \cdot \overline{\boldsymbol{U}}_{\emptyset} = Mm\overline{\boldsymbol{U}}_{\emptyset}.\end{aligned}$$

Thus

$$\sum_{v\in V} \overline{\boldsymbol{U}}_{\{v\}} = \sum_{v\in L_\Phi\cup R_\Phi\cup Z_r} \overline{\boldsymbol{U}}_{\{v\}} = \sum_{(C_i,\alpha)\in L_\Phi} \overline{\boldsymbol{U}}_{\{(C_i,\alpha)\}} + \sum_{(x_j,\alpha,t)\in R_\Phi} \overline{\boldsymbol{U}}_{\{(x_j,\alpha,t)\}} = (M+1)m\overline{\boldsymbol{U}}_{\emptyset}.$$

Now, we calculate the value of the solution

$$\begin{aligned}&\sum_{(u,v)\in\mathcal{E}_\Phi} \left\|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v\}}\right\|^2 \\ =&\sum_{i=1}^{m} \sum_{\alpha:\{x_{i_1},x_{i_2},x_{i_3}\}\to\{0,1\},C_i(\alpha)=1} \sum_{z=1}^{3} \left\|\overline{\boldsymbol{U}}_{\{(C_i,\alpha)\}} - \overline{\boldsymbol{U}}_{\{(x_{i_z},\alpha_{|\{x_{i_z}\}},1)\}}\right\|^2 \\ &+ \sum_{i=1}^{m} \sum_{\alpha:\{x_{i_1},x_{i_2},x_{i_3}\}\to\{0,1\},C_i(\alpha)=1} \sum_{v\in Z_r:((C_i,\alpha),v)\in\mathcal{E}_\Phi} \left\|\overline{\boldsymbol{U}}_{\{(C_i,\alpha)\}} - \overline{\boldsymbol{U}}_{\{v\}}\right\|^2\end{aligned}$$

$$
\begin{aligned}
&+\sum_{j=1}^{n}\sum_{\alpha:\{x_j\}\to\{0,1\}}\sum_{z_1,z_2\in[M\beta]}\left\|\overline{\boldsymbol{U}}_{\{(x_j,\alpha,z_1)\}}-\overline{\boldsymbol{U}}_{\{(x_j,\alpha,z_2)\}}\right\|^2+\sum_{v_1,v_2\in Z_r}\left\|\overline{\boldsymbol{U}}_{\{v_1\}}-\overline{\boldsymbol{U}}_{\{v_2\}}\right\|^2\\
=&\sum_{i=1}^{m}\sum_{\alpha:\{x_{i_1},x_{i_2},x_{i_3}\}\to\{0,1\},C_i(\alpha)=1}\left(\sum_{z=1}^{3}\left\|\overline{\boldsymbol{U}}_{\{(C_i,\alpha)\}}-\overline{\boldsymbol{U}}_{\{(x_{i_z},\alpha_{|\{x_{i_z}\}},1)\}}\right\|^2+2\left\|\overline{\boldsymbol{U}}_{\{(C_i,\alpha)\}}\right\|^2\right)\\
=&\sum_{i=1}^{m}\sum_{\alpha:\{x_{i_1},x_{i_2},x_{i_3}\}\to\{0,1\},C_i(\alpha)=1}\left(\sum_{z=1}^{3}\left\|\boldsymbol{W}_{(\{x_{i_1},x_{i_2},x_{i_3}\},\alpha)}-\boldsymbol{W}_{(\{x_{i_z}\},\alpha_{|\{x_{i_z}\}})}\right\|^2\right.\\
&\left.\qquad+2\left\|\boldsymbol{W}_{(\{x_{i_1},x_{i_2},x_{i_3}\},\alpha)}\right\|^2\right)\\
=&\sum_{i=1}^{m}\sum_{\alpha:\{x_{i_1},x_{i_2},x_{i_3}\}\to\{0,1\},C_i(\alpha)=1}\left(\sum_{z=1}^{3}\left(\left\|\boldsymbol{W}_{(\{x_{i_z}\},\alpha_{|\{x_{i_z}\}})}\right\|^2-\left\|\boldsymbol{W}_{(\{x_{i_1},x_{i_2},x_{i_3}\},\alpha)}\right\|^2\right)\right.\\
&\left.\qquad+2\left\|\boldsymbol{W}_{(\{x_{i_1},x_{i_2},x_{i_3}\},\alpha)}\right\|^2\right)\\
=&\sum_{i=1}^{m}\sum_{\alpha:\{x_{i_1},x_{i_2},x_{i_3}\}\to\{0,1\},C_i(\alpha)=1}\sum_{z=1}^{3}\left\|\boldsymbol{W}_{(\{x_{i_z}\},\alpha_{|\{x_{i_z}\}})}\right\|^2-\\
&\qquad\sum_{i=1}^{m}\sum_{\alpha:\{x_{i_1},x_{i_2},x_{i_3}\}\to\{0,1\},C_i(\alpha)=1}\left\|\boldsymbol{W}_{(\{x_{i_1},x_{i_2},x_{i_3}\},\alpha)}\right\|^2\\
=&\sum_{i=1}^{m}\sum_{z=1}^{3}2\left(\left\|\boldsymbol{W}_{\{x_{i_z}\},\{x_{i_z}\to 0\}}\right\|^2+\left\|\boldsymbol{W}_{\{x_{i_z}\},\{x_{i_z}\to 1\}}\right\|^2\right)-m\\
=&6m-m=5m.
\end{aligned}
$$

□

6.3.3 Soundness : bound for integral solutions

Let $\mathcal{L}=\{(x_j,\alpha)|\alpha:\{x_j\}\to\{0,1\}\}$ be the set of $2n$ literals. For each literal $(x_j,\alpha)\in\mathcal{L}$, let $\deg((x_j,\alpha))$ be the number of left vertices that connect to the literal's representative vertex $(x_j,\alpha,1)$. For a set of literals $\mathcal{L}'\subseteq\mathcal{L}$, let $\deg(\mathcal{L}')=\sum_{(x_j,\alpha)\in\mathcal{L}'}\deg((x_j,\alpha))$. Also, given a subset $\mathcal{L}'\subseteq\mathcal{L}$, for left vertex (C_i,α), say (C_i,α) is *contained* in $\mathcal{L}'$ if all the three literals corresponding to the three neighbors of (C_i, α) in H_Φ are contained in $\mathcal{L}'$, i.e.

$$\{(x_{i_1}, \alpha_{|x_{i_1}}), (x_{i_2}, \alpha_{|x_{i_2}}), (x_{i_3}, \alpha_{|x_{i_3}})\} \subseteq \mathcal{L}'.$$

We first prove the following lemma regarding the structure of $\mathcal{H}_\Phi$, defined by a random 3-XOR instance Φ.

Lemma 6.3.3. *Over the choice of random* 3-XOR *instance* Φ*, with probability* $1 - o(1)$*, the following statements hold.*

- *For each* $\mathcal{L}' \subseteq \mathcal{L}$*,* $|\mathcal{L}'| \geq n/3$*, we have* $\deg(\mathcal{L}') \geq 6m \cdot |\mathcal{L}'|/n(1 - 20/\sqrt{\beta})$.
- *For each* $\mathcal{L}' \subseteq \mathcal{L}$*,* $|\mathcal{L}'| \geq n/3$*, the number of left vertices in* L_Φ *contained in* $\mathcal{L}'$ *is at most* $m \cdot |\mathcal{L}'|^3/(2n^3) \cdot (1 + 100/\sqrt{\beta})$.

Proof. Fix a literal (x_j, α), a random constraint C_i accesses x_j with probability $3/n$. Once C_i accesses x_j, there are 2 vertices out of the 4 left vertices corresponding to C_i adjacent to (x_j, α). Therefore, in expectation, there are $6/n$ edges from the left vertices corresponding to C_i to (x_j, α). By linearity of expectation, for fixed $\mathcal{L}' \subseteq \mathcal{L}$, there are $6|\mathcal{L}'|/n$ edges from the left vertices corresponding to a random constraint C_i to $\mathcal{L}'$ in expectation.

Now for each C_i, let the random variable X_i be the number of representative vertices in $\mathcal{L}'$ that is connected to left vertices corresponding to C_i. By definition we have $\deg(\mathcal{L}') = \sum_{i=1}^m X_i$. Since each left vertex corresponding to C_i has 3 neighbors on the right side, and there are 4 of such left vertices, we know that $X_i \in [0, 12]$. In the previous paragraph we have concluded that $\mathbf{E}[X_i] = 6|\mathcal{L}'|/n$ for all $i = 1, 2, \ldots, m$. It is also easy to see that $X_1, X_2, \ldots, X_m$ are independent random variables.

Now assuming that $|\mathcal{L}'| \geq n/3$, we use Hoeffding's inequality for the random variables $X_1, X_2, \ldots, X_m$, and get

$$\Pr[\deg(\mathcal{L}') < 6m \cdot |\mathcal{L}'|/n(1 - 20/\sqrt{\beta})] = \Pr\left[\sum_{i=1}^{n} X_i < 6m \cdot |\mathcal{L}'|/n(1 - 20/\sqrt{\beta})\right]$$
$$\leq \exp\left(-\frac{2 \cdot \left(\frac{20}{\sqrt{\beta}} \cdot 6m \cdot \frac{|\mathcal{L}'|}{n}\right)^2}{m \cdot 12^2}\right) = \exp\left(-200 \cdot \left(\frac{|\mathcal{L}'|}{n}\right)^2 \cdot n\right) \leq \exp(-22n) \leq 2^{-4n}.$$

Since there are at most 2^{2n} such $\mathcal{L}'$'s, by a union bound, with probability at least $1 - 2^{-2n}$, the first statement holds.

For the second statement, fix an $\mathcal{L}' \subseteq \mathcal{L}$, let a_0, a_1, a_2 be the number of variables that have $0, 1, 2$ corresponding literals in $\mathcal{L}'$, respectively. Note that $a_0 + a_1 + a_2 = n$ and $a_1 + 2a_2 = |\mathcal{L}'|$ Now, for a random constraint C_i, we are interested in the expected number of the four corresponding left vertices (C_i, α) that are contained in $\mathcal{L}'$. Note that once C_i accesses a variable that corresponds to a_0, none of the four corresponding left vertices are contained in $\mathcal{L}'$. Now let us condition on the case that, out of the 3 variables accessed by C_i, t variables have two literals in $\mathcal{L}'$ and the other $(3-t)$ variables have one literal in $\mathcal{L}'$. Observe that in expectation (which is over the random choice of C_i while conditioned on t), there are 2^{t-1} left vertices corresponding to C_i contained in $\mathcal{L}'$.

In all, the expected number of the left vertices corresponding to C_i that are contained in $\mathcal{L}'$ is

$$\sum_{t=0}^{3} \frac{\binom{a_1}{3-t}\binom{a_2}{t}}{\binom{n}{3}} \cdot 2^{t-1} < (1 + \frac{10}{n}) \sum_{t=0}^{3} \binom{3}{t} (a_1/n)^{3-t} (a_2/n)^t \cdot 2^{t-1} \qquad \text{(for } n > 3)$$
$$= (1 + \frac{10}{n})(a_1 + 2a_2)^3/(2n^3) = (1 + \frac{10}{n}) \cdot |\mathcal{L}'|^3/(2n^3).$$

For each C_i, let the random variable X_i be the number of left vertices corresponding to C_i that are contained in $\mathcal{L}'$. By the discuss above, we know that $\mathbf{E}[X_i] < (1 + \frac{10}{n}) \cdot |\mathcal{L}'|^3/(2n^3)$. Now we are interested in the probability that the total number of left vertices contained in $\mathcal{L}'$ (i.e. $\sum_{i=1}^{m} X_i$) is big. Since X_i's are always bounded by $[0, 4]$, by standard Chernoff bound, we have

$$\Pr\left[\sum_{i=1}^{m} X_i > m \cdot |\mathcal{L}'|^3/(2n^3) \cdot (1 + 100/\sqrt{\beta})\right]$$
$$= \Pr\left[\sum_{i=1}^{m} X_i > m \cdot \left(1 + \frac{10}{n}\right) \cdot |\mathcal{L}'|^3/(2n^3) \cdot \frac{1 + 100/\sqrt{\beta}}{1 + 10/n}\right]$$
$$= \Pr\left[\sum_{i=1}^{m} X_i > m \cdot \left(1 + \frac{10}{n}\right) \cdot |\mathcal{L}'|^3/(2n^3) \cdot \left(1 + \frac{100/\sqrt{\beta} - 10/n}{1 + 10/n}\right)\right]$$
$$\le \exp\left(-\frac{1}{4} \cdot m \cdot \left(1 + \frac{10}{n}\right) \cdot |\mathcal{L}'|^3/(2n^3) \cdot \frac{(100/\sqrt{\beta} - 10/n)^2}{3(1 + 10/n)^2}\right) \qquad \text{(for large enough } \beta)$$
$$\le \exp\left(-\frac{1}{4} \cdot m \cdot |\mathcal{L}'|^3/(2n^3) \cdot \frac{(80/\sqrt{\beta})^2}{3}\right) \qquad \text{(for } n \gg \sqrt{\beta} \gg 1)$$
$$= \exp\left(-\beta n \cdot \frac{|\mathcal{L}'|^3}{n^3} \cdot \frac{1}{\beta} \cdot \frac{800}{3}\right)$$
$$\le \exp\left(-n \cdot \frac{800}{3^4}\right) \qquad \text{(since } |\mathcal{L}'| \ge n/3)$$

$\leq 2^{-4n}$.

Since there are at most 2^{2n} such $\mathcal{L}'$'s, by a union bound, with probability at least $1 - 2^{-2n}$, the second statement holds. $\square$

Now, we are ready to prove the soundness lemma.

Lemma 6.3.4 (Soundness)**.** *For $\tau > 1/3$, with probability $1 - o(1)$, the τ vs. $(1 - \tau)$ balanced separator has at least $4m(3\tau - \tau^3 - O(1/\sqrt{\beta}) - O(1/M))$ edges in the cut.*

Proof. We are going to prove that, once the two conditions in Lemma 6.3.3 hold, we have the desired upper bound for τ vs. $(1 - \tau)$ balanced separator. Let us assume that there is a balanced separator (A', B') such that $\mathrm{edges}(A', B') \leq 4m(3\tau - \tau^3) \leq 12m$, we will show that $\mathrm{edges}(A', B') \geq 4m(3\tau - \tau^3 - O(1/\sqrt{\beta}) - O(1/M))$.

Based on (A', B') we build another cut (A, B) such that $A \cap Z_r = A' \cap Z_r$ and $A \cap R_\Phi = A' \cap R_\Phi$. For each left vertex in L_Φ, it has 5 edges going to Z_r and R_Φ. We assign the vertex to A if it has less than 3 edges going to $B' \cap (Z_r \cup R_\Phi)$, and assign it to B otherwise. Note that $\mathrm{edges}(A, B) \leq \mathrm{edges}(A', B')$, therefore we only need to show that $\mathrm{edges}(A, B) \geq m(12\tau - \tau^3 - O(1/\sqrt{\beta}) - O(1/M))$. Since L_Φ contains only $O(1/M)$ fraction of the total vertices, (A, B) is still $(\tau - O(1/M))$ vs. $(1 - \tau + O(1/M))$ balanced.

Since $\mathrm{edges}(A, B) \leq 12m$, for large enough constant M, we have the following two statements.

1) One of $A \cap Z_r$ and $B \cap Z_r$ has at most $100/M \cdot |Z_r| = 100m/M$ vertices.

2) Let $\mathcal{C}_{\mathrm{bad}} = \{(x_j, \alpha) :$ the clique $C_{(x_j,\alpha)}$ is broken by $(A, B)\}$, then $|\mathcal{C}_{\mathrm{bad}}| \leq 20n/M$.

If 1) does not hold, then we see there are at least $(100/M) \cdot |Z_r| \cdot M = 100m$ edges in Z_r cut by (A, B), by the expansion property. If 2) does not hold, for each clique $C_{(x_j,\alpha)}$ that is broken by (A, B), at least $(\beta M - 1)$ edges of the clique are in the cut. In all, there are at least $(\beta M - 1) \cdot 20n/M > 12\beta n = 12m$ edges in the cut.

Now, by 1), assume w.l.o.g. that $A \cap Z_r$ is the smaller side – having at most $100/M \cdot |Z_r|$ vertices, and let $\mathcal{L}'$ be the set of literals (x_j, α) such that its representative vertex $(x_j, \alpha, 1)$ is in A.

To get a lower bound for $|\mathcal{L}'|$, note that

$$|A| \leq (|\mathcal{L}'| + |\mathcal{C}_{\mathrm{bad}}|) \cdot M\beta + |Z_r| + |L_\Phi| = |\mathcal{L}'| \cdot M\beta + O(1)m. \tag{6.5}$$

Also, since (A, B) is a balanced separator, we have $|A| \geq (\tau - O(1/M)) \cdot 2Mm$. Hence, by (6.5), we have $|\mathcal{L}'| \geq (\tau - O(1/M)) \cdot 2n$.

Let $L_{\mathrm{bad}} \subseteq L_\Phi$ be the set of left vertices such that at least one of the two neighbors in Z_r falls into $A \cap Z_r$. By the regularity of the graph where Z_r is the set of vertices and L_Φ is the set of edges, we know that $|L_{\mathrm{bad}}| \leq 8 \cdot 100/M \cdot |Z_r| \leq O(m/M)$.

Now let us get a lower bound on $\mathrm{edges}(A, B)$. First, we have $\mathrm{edges}(A, B) \geq \mathrm{edges}(A \setminus L_{\mathrm{bad}}, B \setminus L_{\mathrm{bad}})$. Let $L'_\Phi = L_\Phi \setminus L_{\mathrm{bad}}$, we have

$$\begin{aligned}
&\mathrm{edges}(A \setminus L_{\mathrm{bad}}, B \setminus L_{\mathrm{bad}}) \\
&= \mathrm{edges}(A \cap (L'_\Phi \cup R_\Phi \cup Z_r), B \cap (L'_\Phi \cup R_\Phi \cup Z_r)) \\
&\geq \mathrm{edges}(A \cap L'_\Phi, B \cap Z_r) + \mathrm{edges}(A \cap R_\Phi, B \cap L'_\Phi) \\
&= \mathrm{edges}(A \cap L'_\Phi, B \cap Z_r) + \mathrm{edges}(A \cap R_\Phi, L'_\Phi) - \mathrm{edges}(A \cap R_\Phi, A \cap L'_\Phi) \\
&\geq \mathrm{edges}(A \cap L'_\Phi, B \cap Z_r) + \mathrm{edges}(A \cap R_\Phi, L_\Phi) - |L_{\mathrm{bad}}| \cdot 3 - \mathrm{edges}(A \cap R_\Phi, A \cap L'_\Phi).
\end{aligned}$$

Consider a left vertex $(C_i, \alpha) \in L'_\Phi$. We claim that it is contained in $\mathcal{L}'$ if and only if $(C_i, \alpha) \in A$. This is because if it is contained in $\mathcal{L}'$, then we have $(C_i, \alpha) \in A$ because 3 out of 5 edges incident to (C_i, α) go to A side (the three variable representative vertices). If (C_i, α) is not contained in $\mathcal{L}'$, we have at least 3 out of the 5 edges going to B side (the two edges to $B \cap Z_r$ and at least one of the variable representative vertices), and therefore we have $(C_i, \alpha) \in B$. By this claim, we know the following two facts.

- $|A \cap L'_\Phi|$ is small. Since $\tau > 1/3$, we have $|\mathcal{L}'| \geq (2/3 - O(1/M))n > n/3$, and by the second property of Lemma 6.3.3, we have $|A \cap L'_\Phi| \leq m \cdot |\mathcal{L}'|^3/(2n^3) \cdot (1 + 100/\sqrt{\beta})$.
- We have $\mathrm{edges}(A \cap L'_\Phi, B \cap Z_r) = 2|A \cap L'_\Phi|$ and $\mathrm{edges}(A \cap L'_\Phi, A \cap R_\Phi) = 3|A \cap L'_\Phi|$.

For $\mathrm{edges}(A \cap R_\Phi, L_\Phi)$, we know that this is exactly $\deg(\mathcal{L}')$. Again, since $\tau > 1/3$, by the first property of Lemma 6.3.3, we know this value is lower-bounded by $6m \cdot |\mathcal{L}'|/n(1 - 20/\sqrt{\beta})$.

In all, we have

$$\begin{aligned}
&\mathrm{edges}(A, B) \\
&\geq \mathrm{edges}(A \cap L'_\Phi, B \cap Z_r) + \mathrm{edges}(A \cap R_\Phi, L_\Phi) - |L_{\mathrm{bad}}| \cdot 3 - \mathrm{edges}(A \cap R_\Phi, A \cap L'_\Phi) \\
&= 2|A \cap L'_\Phi| + \deg(\mathcal{L}') - |L_{\mathrm{bad}}| \cdot 3 - 3|A \cap L'_\Phi| \\
&\geq \deg(\mathcal{L}') - |A \cap L'_\Phi| - O(m/M)
\end{aligned}$$

$$
\begin{aligned}
&\geq 6m \cdot |\mathcal{L}'|/n(1-20/\sqrt{\beta}) - m \cdot |\mathcal{L}'|^3/(2n^3) \cdot (1+100/\sqrt{\beta}) - O(m/M) \\
&= m\left(12\gamma - 4\gamma^3 - (240\gamma + 400\gamma^3)/\sqrt{\beta} - O(1/M)\right) \qquad\qquad (\text{let } \gamma = |\mathcal{L}'|/(2n)) \\
&\geq 4m\left(3\tau - \tau^3 - O(1/\sqrt{\beta}) - O(1/M)\right).
\end{aligned}
$$

The last step follows because (i) $3\gamma - \gamma^3$ monotonically increases when $\gamma \in [0,1]$, and (ii) $\gamma \geq (\tau - O(1/M))$. □

6.3.4 Constant-degree integrality gap instance

In this subsection, we slightly modify the graph $\mathcal{H}_\Phi$ obtained in the previous subsections to get an integrality gap instance with constant degree.

Observe that in $\mathcal{H}_\Phi$, when M and β are constants, the only vertices whose degree might be superconstant are the representative vertices in R_Φ. Now consider the edges connecting vertices in L_Φ and representative vertices: there are $12m$ of them, each of them corresponds to a combination of constraint C_i, satisfying assignment α, and one of the variables in the constraint. Let E_b be the set of these edges.

For two edges $e_1, e_2 \in E_b$, let the random variable $Y_{\{e_1,e_2\}} = 1$ if they share the same representative vertex, and let $Y_{\{e_1,e_2\}} = 0$ otherwise. Finally let $Y = \sum_{e_1,e_2 \in E_b} Y_{\{e_1,e_2\}}$. By the simple second moment method, we know that with probability $1 - o(1)$, we have $Y \leq \frac{1000m^2}{n} = 1000\beta^2 n$.

For every edge $e \in E_b$, if $\sum_{e' \in E_b \setminus \{e\}} Y_{\{e,e'\}} > \beta M$, we remove e from the graph. In this way, we get a new graph, namely $\mathcal{H}'_\Phi$. We claim the following properties about $\mathcal{H}'_\Phi$.

1. The maximum degree of $\mathcal{H}'_\Phi$ is $O(\beta M)$. This is because the maximum degree of vertices other than representative vertices in $\mathcal{H}_\Phi$ is $O(\beta M)$, and after the edge removal process described above, the representative vertices have degree $O(\beta M)$.

2. The number of edges removed is at most $2Y/(\beta M)$, and therefore $2000m/M$ with probability $1 - o(1)$. This is because whenever an edge is removed, we charge βM to Y. Since each edge in Y can be charge at most twice, there are at most $2Y/(\beta M)$ edges to be removed.

3. The SDP solution in Lemma 6.3.2 is still feasible and has objective value at most $5m$ (since we removed edges) with probability $1 - o(1)$.

4. The soundness lemma Lemma 6.3.4 still holds since we removed only $O(m/M)$ edges.

Therefore, we claim that $\mathcal{H}'_\Phi$ is an integrality gap instance for Theorem 6.3.1 with constant degree.

6.4 Gaps for UNIFORMSPARSESTCUT

In this section, we provide the full analysis of the gap instance for UNIFORMSPARSEST-CUT. We first describe our construction of the gap instance for UNIFORMSPARSESTCUT as follows.

We modify the gap instance we got for BALANCEDSEPARATOR to get an instance for the linear round Lasserre relaxation of UNIFORMSPARSESTCUT. The reduction converts the gap instance for BALANCEDSEPARATOR to the gap instance for UNIFORMSPARSES-TCUT in an almost black box style. In the BALANCEDSEPARATOR problem, we have the hard constraint that the cut is τ-balanced. In the reduction from BALANCEDSEPARATOR to UNIFORMSPARSESTCUT, we need to use the sparsity objective to enforce this constraint. We do it as follows. Recall that given a 3-XOR instance Φ, the corresponding gap instance for BALANCEDSEPARATOR consists of vertex set $L_\Phi \cup R_\Phi \cup Z_r$ and edge set $\mathcal{E}_\Phi$. To get a gap instance for UNIFORMSPARSESTCUT, we add two more $O(M)$-regular expanders (with edge expansion $10^4 \cdot M$) D_l and D_r of size $1000Mm$ (where M is the same parameter defined in the previous sections). Now, let the edge set $\mathcal{E}'_\Phi$ contain the edges in $\mathcal{E}_\Phi$, in the expanders D_l and D_r, and the following edges : for each vertex $v \in L_\Phi \cup R_\Phi \cup Z_r$, introduce 2 new edges incident to it, one to a vertex in D_l (say, v_l) and the other one to a vertex in D_r (say, v_r). We arrange these edges (between $L_\Phi \cup R_\Phi \cup Z_r$ and D_l, D_r) in a way so that each vertex in D_l (or D_r) has at most one neighbor in $L_\Phi \cup R_\Phi \cup Z_r$ – this can be done because $|L_\Phi| + |R_\Phi| + |Z_r| = (2M+5)m < 1000Mm = |D_l| = |D_r|$.

Using the instance described above, we will prove our main integrality gap theorem (Theorem 6.1.3) for UNIFORMSPARSESTCUT. We state the full theorem as follows.

Theorem 6.4.1. *For large enough constant β, M (where β is the same parameter as in previous sections), and infinitely many positive integer N's, there is an N-vertex instance for* UNIFORMSPARSESTCUT *problem, such that the optimal solution is at least $(1 + 1/(100M))$ times worse than the optimal solution of the $\Omega(N)$-round Lasserre SDP.*

Theorem 6.4.1 is directly implied by the following completeness lemma (Lemma 6.4.2) and soundness lemma (Lemma 6.4.3).

Lemma 6.4.2 (Completeness)**.** *The value of relaxation Ψ_2 (in Figure 6.2) is at most*

$$(2M+10)m/((1001M+1)m)^2$$

for $\tau = (1001M+1)/(2002M+5)$.

Proof. Given the SDP solution $\{\overline{\boldsymbol{U}}_{S'}\}_{S' \subseteq L_\Phi \cup R_\Phi \cup Z_r, |S'| \le r/3}$ in the completeness case of BALANCEDSEPARATOR, we extend it to the SDP solution $\{\overline{\boldsymbol{U}}_S\}_{S \subseteq L_\Phi \cup R_\Phi \cup Z_r \cup D_l \cup D_r, |S| \le r/3}$ for UNIFORMSPARSESTCUT by "putting D_l and D_r one per side". That is, for each $S \subseteq L_\Phi \cup R_\Phi \cup Z_r \cup D_l \cup D_r$ with $|S| \le r/3$, let $S' = S \cap (L_\Phi \cup R_\Phi \cup Z_r)$. Now we let $\overline{\boldsymbol{U}}_S = \mathbf{0}$ if $S \cap D_r \ne \emptyset$, and let $\overline{\boldsymbol{U}}_S = \overline{\boldsymbol{U}}_{S'}$ otherwise.

We first check that $\{\overline{\boldsymbol{U}}_S\}_{S \subseteq L_\Phi \cup R_\Phi \cup Z_r \cup D_l \cup D_r, |S| \le r/3}$ is a feasible SDP solution. We only check that the balance constraint (the last constraint in relaxation Φ_2) is met.

We are going to prove prove that

$$\sum_{u \in L_\Phi \cup R_\Phi \cup Z_r \cup D_l \cup D_r} \overline{\boldsymbol{U}}_{\{u\}} = (1001M+1)m\overline{\boldsymbol{U}}_\emptyset.$$

From the proof of Lemma 6.3.2, we know that

$$\sum_{u \in L_\Phi \cup R_\Phi \cup Z_r} \overline{\boldsymbol{U}}_{\{u\}} = (M+1)m\overline{\boldsymbol{U}}_\emptyset,$$

together with the fact that

$$\forall u \in D_l, \overline{\boldsymbol{U}}_{\{u\}} = \overline{\boldsymbol{U}}_\emptyset, \qquad \forall u \in D_r, \overline{\boldsymbol{U}}_{\{u\}} = \mathbf{0},$$

we get the desired equality.

Now we calculate the value of the solution. First, we calculate the following value.

$$\begin{aligned}
&\sum_{(u,v)\in\mathcal{E}'_\Phi} \left\|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v\}}\right\|^2 = \sum_{(u,v)\in\mathcal{E}_\Phi} \left\|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v\}}\right\|^2 + \sum_{(u,v)\in\mathcal{E}'_\Phi\setminus\mathcal{E}_\Phi} \left\|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v\}}\right\|^2 \\
=&5m + \sum_{u,v\in D_l} \left\|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v\}}\right\|^2 + \sum_{u,v\in D_r} \left\|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v\}}\right\|^2 \\
&\qquad + \sum_{u\in L_\Phi\cup R_\Phi\cup Z_r} \left(\left\|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v_l\}}\right\|^2 + \left\|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v_r\}}\right\|^2\right),
\end{aligned}$$

Note that $\sum_{u,v\in D_l} \left\|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v\}}\right\|^2 + \sum_{u,v\in D_r} \left\|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v\}}\right\|^2 = 0$, and

$$\sum_{u\in L_\Phi\cup R_\Phi\cup Z_r} \left(\left\|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v_l\}}\right\|^2 + \left\|\overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v_r\}}\right\|^2\right)$$

$$= \sum_{u \in L_\Phi \cup R_\Phi \cup Z_r} \left(2 \left\| \overline{\boldsymbol{U}}_{\{u\}} \right\|^2 + \left\| \overline{\boldsymbol{U}}_{\{v_l\}} \right\|^2 + \left\| \overline{\boldsymbol{U}}_{\{v_r\}} \right\|^2 - 2\langle \overline{\boldsymbol{U}}_{\{u\}}, \overline{\boldsymbol{U}}_{\{v_l\}} \rangle - 2\langle \overline{\boldsymbol{U}}_{\{u\}}, \overline{\boldsymbol{U}}_{\{v_r\}} \rangle \right)$$

$$= \sum_{u \in L_\Phi \cup R_\Phi \cup Z_r} \left(2 \left\| \overline{\boldsymbol{U}}_{\{u\}} \right\|^2 + 1 + 0 - 2 \left\| \overline{\boldsymbol{U}}_{\{u,v_l\}} \right\|^2 - 2 \left\| \overline{\boldsymbol{U}}_{\{u,v_r\}} \right\|^2 \right)$$

(by property of Lasserre vectors)

$$= \sum_{u \in L_\Phi \cup R_\Phi \cup Z_r} 1 = |L_\Phi| + |R_\Phi| + |Z_r| = (2M+5)m.$$

Thus, we have

$$\sum_{(u,v) \in \mathcal{E}'_\Phi} \left\| \overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v\}} \right\|^2 = (2M+10)m.$$

Since $\tau < 1/2$, the value of the solution is at most

$$\frac{1}{|L_\Phi \cup R_\Phi \cup Z_r \cup D_l \cup D_r|^2 \tau^2} \sum_{(u,v) \in \mathcal{E}'_\Phi} \left\| \overline{\boldsymbol{U}}_{\{u\}} - \overline{\boldsymbol{U}}_{\{v\}} \right\|^2 = (2M+10)m/((1001M+1)m)^2.$$

□

Lemma 6.4.3 (Soundness)**.** *For large enough M, the sparsity of the sparsest cut is at least* $\gamma = (1 + 1/(100M)) \cdot (2M+10)m/(1001Mm)^2$.

Proof. Let D'_l be the smaller part among $D_l \cap S$ and $D_l \cap \bar{S}$, and D''_l be the larger part. Also, let D'_r be the smaller part among $D_r \cap S$ and $D_r \cap \bar{S}$ and D''_r be the larger part. Let $(T, \bar{T})$ be the cut restricted to $L_\Phi \cup R_\Phi \cup Z_r$ (the BALANCEDSEPARATOR instance), i.e. let $T = S \cap (L_\Phi \cup R_\Phi \cup Z_r)$ and $\bar{T} = \bar{S} \cap (L_\Phi \cup R_\Phi \cup Z_r)$.

First, we show that to get a cut of sparsity better than γ, $|D'_l| \leq \frac{1}{10^4 M} \cdot |D_l|$, and the same is true for D_r (by the same argument). This is because if $|D'_l| > \frac{1}{10^4 M} \cdot |D_l|$, by the expansion property, there are at least $10^4 M \cdot |D'_l| > 1000Mm$ edges in the cut. Since the graph has $|L_\Phi| + |R_\Phi| + |Z_r| + |D_l| + |D_r| = (2002M+5)m$ vertices, therefore the sparsity of the cut is at least

$$\frac{1000Mm}{\frac{1}{4} \cdot ((2002M+5)m)^2} > \frac{500Mm}{(1001Mm)^2} > \gamma,$$

for $M > 1/25$.

Second, we show that D'_l and D'_r should be on opposite sides of any cut of sparsity better than γ. Suppose not, let S be the side of the cut which D'_l and D'_r are on. Recall that $T = S \cap (L_\Phi \cup R_\Phi \cup Z_r)$. We have

$$\mathrm{edges}(S, \bar{S}) \geq \mathrm{edges}(T, D''_l \cup D''_r) + \mathrm{edges}(D'_l, D''_l) + \mathrm{edges}(D'_r, D''_r).$$

Note that $\mathrm{edges}(T, D''_l \cup D''_r) \geq 2|T| - |D'_l| - |D'_r|$ as each vertex in D_l, D_r is connected to at most one vertex in T. Also, by the expansion property, $\mathrm{edges}(D'_l, D''_l) + \mathrm{edges}(D'_r, D''_r) \geq 1000M(|D'_l| + |D'_r|)$. Now, we have

$$\begin{aligned}\mathrm{edges}(S, \bar{S}) &\geq 2|T| - (|D'_l| + |D'_r|) + 1000M(|D'_l| + |D'_r|)\\ &= 2(|T| + |D'_l| + |D'_r|) + (1000M - 3)(|D'_l| + |D'_r|) \geq 2(|T| + |D'_l| + |D'_r|) = 2|S|.\end{aligned}$$

Therefore, the sparsity of the cut

$$\frac{\mathrm{edges}(S, \bar{S})}{|S||\bar{S}|} \geq \frac{2|S|}{|S||\bar{S}|} = \frac{2}{|\bar{S}|} \geq \frac{2}{(2002M + 5)m} > \gamma.$$

Third, we show that if the cut $(S, \bar{S})$ has sparsity better than γ, then the cut $(T, \bar{T})$ defined above is a 0.49 vs 0.51 balanced cut, i.e. $|T|/(|L_\Phi| + |R_\Phi| + |Z_r|) \in [0.49, 0.51]$. Supposing $(T, \bar{T})$ is not 0.49 vs 0.51 balanced, i.e. $||T| - |\bar{T}|| > 0.02 \cdot (2M + 5)m$, we have

$$\begin{aligned}||S| - |\bar{S}|| \geq ||T| - |\bar{T}|| - |D'_l| - |D'_r| &\geq 0.02 \cdot (2M + 5)m - \frac{2}{1000M} \cdot 1000Mm\\ &\geq (0.04M - 2)m \geq 0.01Mm,\end{aligned}$$

for large enough M. Therefore, $(S, \bar{S})$ is not $0.5 - 10^{-6}$ vs $0.5 + 10^{-6}$ balanced. Thus,

$$|S||\bar{S}| < ((2002M + 5)m)^2 \cdot (0.5 - 10^{-6})(0.5 + 10^{-6}) < (1001Mm)^2 \cdot (1 - 10^{-12}).$$

Since D''_l and D''_r are on opposite sides of $(S, \bar{S})$, we know that $\mathrm{edges}(S, \bar{S}) \geq (2M + 5)m - |D'_l| - |D'_r| \geq (2M + 5)m \cdot (1 - 1/M)$, and therefore the sparsity of the cut

$$\frac{\mathrm{edges}(S, \bar{S})}{|S||\bar{S}|} > \frac{(2M + 5)m}{(1001Mm)^2} \cdot (1 - 1/M)(1 + 10^{-12}).$$

This value is greater than γ when $M > 10^{20}$.

Finally, since $(T, \bar{T})$ is a 0.49 vs 0.51 balanced cut, by Lemma 6.3.4, we know that with probability $1 - o(1)$, $\mathrm{edges}(T, \bar{T}) > (5.4 - O(1/\sqrt{\beta}) - O(1/M))m$. Therefore

$$\frac{\mathrm{edges}(S, \bar{S})}{|S||\bar{S}|}$$

$$
\begin{aligned}
&\geq \frac{\text{edges}(T,\bar{T}) + (2M+5)m - |D'_l| - |D'_r|}{\frac{1}{4}\cdot((2002M+5)m)^2} \\
&\geq \frac{(5.4 - O(1/\sqrt{\beta}) - O(1/M))m + (2M+5)m - \frac{1000Mm}{10^4 M} - \frac{1000Mm}{10^4 M}}{\frac{1}{4}\cdot((2002M+5)m)^2} \\
&= \frac{(2M + 10.2 - O(1/\sqrt{\beta}) - O(1/M))m}{\frac{1}{4}\cdot((2002M+5)m)^2} \\
&\geq \frac{(2M + 10.2 - O(1/\sqrt{\beta}) - O(1/M))m}{(1001Mm)^2}\cdot(1 - 1/(200M)) \\
&\geq \frac{(2M+10.1)m}{(1001Mm)^2}\cdot(1-1/(200M)) && \text{(for large enough } \beta \text{ and } M\text{)} \\
&\geq \frac{(2M+10)m}{(1001Mm)^2}\cdot(1+1/(30M))(1-1/(200M)) && \text{(for large enough } M\text{)} \\
&\geq \frac{(2M+10)m}{(1001Mm)^2}\cdot(1+1/(100M)) = \gamma.
\end{aligned}
$$

□

Chapter 7

Lasserre integrality gaps for ROBUSTGISO

7.1 Introduction

The GRAPHISOMORPHISM problem is one of the most intriguing and notorious problems in computational complexity theory (we will also refer to it as GISO for short); we refer to [149, 30, 24, 148, 69] for surveys. It was famously referred to as a "disease" over 35 years ago [199] and maintains its infectious status to this day. Together with FACTORING, it is one of the very rare problems in NP which is not known to be in P but which is believed to not be NP-hard [29, 49, 205] (according to standard complexity-theoretic assumptions). Both problems also admit an algorithm with running time "subexponential" (or "moderately exponential") in the natural witness size. In the case of GRAPHISOMORPHISM on n-vertex graphs, the natural witness size is $\log_2(n!) = \Theta(n \log n)$, but the best known algorithm due to Luks solves the problem in time $2^{O(\sqrt{n \log n})}$ [33].

In the same breath we might mention the problems GAPSVP$_{\sqrt{n}}$ (approximating the shortest vector in an n-dimensional lattice to factor $\sqrt{n}$) and UNIQUEGAMES$_\epsilon$ (the Unique Games problem proposed by Khot [136]). The former is not NP-hard subject to standard complexity-theoretic assumptions [95, 4], though we don't know any subexponential-time algorithm. The latter has a subexponential-time algorithm [16]; whether it is NP-hard or in P (or neither) is hotly contested. The potential hardness of FACTORING and GAPSVP — even under certain average-case distributions — is well enough entrenched that many cryptographic protocols are based on it. (The same is true of random 3-XOR with noise, more on which later.) On the other hand, for GRAPHISOMORPHISM and UNIQUEGAMES$_\epsilon$

we do not know any way of generating "hard-seeming instances"; indeed, some experts have speculated that GRAPHISOMORPHISM may be in P, or at least have a $2^{\mathrm{polylog}(n)}$-time algorithm.

In this chapter we investigate hardness results for the GISO problem. Since GISO may well be in P, let us discuss what this may mean. One direction would be to show that GISO is hard for small complexity classes. This has been pursued most successfully by Torán [217], who has shown that GISO is hard for the class DET. This is essentially the class of problems equivalent to computing the determinant; it contains $\#\mathsf{L}$ and is contained in TC^1. It is not known whether GISO is P-hard.

7.1.1 Our contributions

In this chapter, however, we are concerned with hardness results well above P. Our main contribution is that solving GISO via the Lasserre/SOS hierarchy requires $2^{\Omega(n)}$ time (i.e., $\Omega(n)$ rounds/degree). This generalizes the result of Cai, Fürer, and Immerman [55] showing that the frequently effective $o(n)$-dimensional Weisfeiler–Lehman algorithm fails to solve GISO; it also gives even more evidence that any subexponential-time algorithm for GISO requires algebraic, non-local techniques.

Another result among our contributions is concerned with the problem of *robust* graph isomorphism, ROBUSTGISO. Recall from the introduction part of this the is that ROBUSTGISO is the following problem: given two graphs which are almost isomorphic, find an "almost-isomorphism". ROBUSTGISO is strictly harder than GISO and the fact that it concerns "isomorphisms with noise" seems to rule out all algebraic techniques. We show that ROBUSTGISO is at least as hard as random 3-XOR with noise; hence ROBUSTGISO has no polynomial-time algorithm assuming the well-known R3XOR Hypothesis of Feige [88]. In fact, it's possible that the R3XOR problem requires $2^{n^{1-o(1)}}$ time, which would mean that ROBUSTGISO is much harder than GISO itself.

7.1.2 SOS/Lasserre gaps

The most well-known heuristic for GRAPHISOMORPHISM (and the basis of most practical algorithms — e.g., "nauty" [171]) is the Weisfeiler–Lehman (WL) algorithm [222] and its "higher dimensional" generalizations. To describe the basic algorithm we need the notion of a colored graph. This is simply a graph, together with a function mapping the vertices to a finite set of colors; equivalently, a graph with its vertices partitioned into "color classes". Isomorphisms involving colored graphs are always assumed to preserve colors. Let G be a colored graph on the n-vertex set V. A *color refinement step* refers to the following procedure: for each $v \in V$, one determines the multiset C_v of colors in the neighborhood of v; then one recolors each v with color C_v. Now the basic WL algorithm, when given graphs G and H, repeatedly applies refinement to each of them until the colorings stabilize. (Initially, the graphs are treated as having just one color class.) At the end, if G and H have the same number of vertices of each color the WL algorithm outputs that they are "maybe isomorphic"; otherwise, it (correctly) outputs that they are "not isomorphic".

Note that after the initial refinement step, a graph's vertices are colored according to their degree. Thus two d-regular graphs are always reported as "maybe isomorphic" by the basic WL algorithm. On the other hand, the heuristic is powerful enough to work correctly for all trees and for almost all n-vertex graphs in the Erdős–Rényi $\mathcal{G}(n, 1/2)$ model [31, 32]. (We say the heuristic "works correctly" on a graph G if the stabilized coloring for G is distinct from the stabilized coloring of any graph not isomorphic to G.) To overcome WL's failure for regular graphs, several researchers independently introduced the "k-dimensional generalization" WL^k (see [222, 55] for discussion). Briefly, in the WL^k heuristic, each k-tuple of vertices has a color, and color refinement involves looking at all "neighbors" of each k-tuple of vertices $(v_1, \dots, v_k)$ (where the neighbors are all tuples of the form $(v_1, \dots, v_{i-1}, u_i, v_{i+1}, \dots, v_k)$, where $\{u_i, v_i\}$ is an edge). The WL^k heuristic can be performed in time $n^{k+O(1)}$ and is thus a polynomial-time algorithm for any constant k.

The WL^k heuristic is very powerful. For example, it is known to work correctly in polynomial time for all graphs which exclude a fixed minor [102], a class which includes all graphs of bounded treewidth or bounded genus. Spielman's $2^{\widetilde{O}(n^{1/3})}$-time graph isomorphism algorithm [213] for strongly regular graphs is achieved by WL^k with $k = \widetilde{O}(n^{1/3})$. The WL^k algorithm with $k = O(\sqrt{n})$ is also a key component in the $2^{O(\sqrt{n \log n})}$-time GISO algorithm [33]. Throughout the '80s there was some speculation that GISO might be solvable on all graphs by running the WL^k algorithm with $k = O(\log n)$ or even $k = O(1)$. However this was disproved in the notable work of Cai, Fürer, and Immerman [55], which showed the existence of nonisomorphic n-vertex graphs G and H which are not distinguished by WL^k unless $k = \Omega(n)$.[1]

The GRAPHISOMORPHISM problem can be thought of as kind of constraint satisfaction problem (CSP), and readers familiar with LP/SDP hierarchies for CSPs might see an analogy between k-dimensional WL and level-k LP/SDP relaxations. A very interesting recent work of Atserias and Maneva [27] (see also [103]) shows that this is more than

[1]Actually, G and H are colored graphs in [55]'s construction, with each color class having size at most 4. It is often stated that the colors can be replaced by gadgets while keeping the number of vertices $O(n)$. We do not find this to be immediately obvious. However it does follow from the asymmetry of random graphs, as we will see later in this chapter.

just an analogy — it shows that the power of WL^k is precisely sandwiched between that of the kth and $(k+1)$st level of the canonical *Sherali–Adams LP hierarchy* [207]. (In fact, it had long been known [196] that WL^1 is equivalent in power to the basic LP relaxation of GISO.) This gives a very satisfactory connection between standard techniques in optimization algorithms and the best known non-algebraic/local techniques for GISO.

This connection raises the question of whether stronger LP/SDP hierarchies might prove more powerful than WL^k in the context of GISO. The strongest such hierarchy known is the "SOS (sum-of-squares) hierarchy" due to Lasserre [156] and Parrilo [185]. Very recent work [34, 184, 133] (part of which included in this thesis) in the field of CSP approximability has shown that $O(1)$ levels of the SOS hierarchy can succeed where $\omega(1)$ levels of weaker SDP hierarchies fail; in particular, this holds for the hardest known instances of $\text{UNIQUEGAMES}_\epsilon$ [34]. This raises the question of whether there might be a subexponential-time algorithm based SOS which solves GRAPHISOMORPHISM.

We answer this question negatively. Our first main result is that a variant of the Cai–Fürer–Immerman instances also fools $\Omega(n)$ levels of the SOS hierarchy. In fact, we achieve a "constant factor Lasserre gap with perfect completeness". To explain this, recall the definition of α-isomorphism from Definition 2.1.5 which we restate as follows.

Definition 7.1.1 (Re-statement of Definition 2.1.5)**.** *Let G and H be nonempty n-vertex graphs. For $0 \le \beta \le 1$, we say that a bijection $\pi : V(G) \to V(H)$ is an α-*isomorphism *if*

$$\frac{|\{(u,v) \in E(G) : (\pi(u),\pi(v)) \in E(H)\}|}{\max\{|E(G)|,|E(H)|\}} \ge \alpha.$$

*In this case we say that G and H are α-*isomorphic.

Observe that this definition is symmetric in G and H. The two graphs are isomorphic if and only if they are 1-isomorphic. We will almost always consider the case that G and H have the same number of edges. We prove:

Theorem 7.1.2. *For infinitely many n, there exist pairs of n-vertex, $O(n)$-edge graphs G and H such that:*

- *G and H are not $(1-10^{-14})$-isomorphic;*
- *any SOS refutation of the statement "G and H are isomorphic"[2] requires degree $\Omega(n)$.*

[2]When naturally encoded.

A word on our techniques. The essence of the Cai–Fürer–Immerman construction is to take a 3-regular expander graph and replace each vertex by a certain 10-vertex gadget (originally appearing in [123] and also sometimes called a "Fürer gadget"). This gadget is closely related to 3-variable equations modulo 2 (as observed by several authors, e.g. [217]); indeed, it may be described as the "label-extended graph" of the 3-XOR constraint. The reader may therefore recognize the [55] WL^k lower bound as stemming from the difficulty of refuting unsatisfiable, expanding 3-XOR CSP instances by "local" means. This should make our Theorem 7.1.2 look plausible in light of the Grigoriev [99] and Schoenebeck [204] SOS/Lasserre lower bounds.

Nevertheless, obtaining Theorem 7.1.2 is not automatic. For one, we still lack a complete theory of reductions within the SOS hierarchy (though see [220]). Second, the pair of graphs constructed by [55] only differ by one edge. More tricky is the issue of removing the "colors" from the [55] construction. We do not see an easy gadget-based way of doing this without sacrificing on the $\Omega(n)$ degree. To handle this we have to: a) modify the [55] construction somewhat to make the two graphs differ by a constant fraction; b) prove that random (hyper)graphs are "robustly asymmetric" — i.e., "far from having nontrivial automorphisms"; c) use the robust asymmetry property to remove the "colors". The result in b), described below in Section 7.1.4, qualitatively generalizes work of Erdős and Rényi [81] and may be of independent interest.

Comparison to the work by Snook et al. In an independent work, Schoenebeck, Codenotti, and Snook [211] have shown a conclusion similar to our Theorem 7.1.2. Their main result is that there are expander graphs G and H which are not isomorphic, but any SOS refutation of the statement "G and H are isomorphic" requires degree $\Omega(n)$. As in our work, their proof combines the Cai–Fürer–Immerman construction with the Schoenebeck [204] SOS/Lasserre lower bounds.

7.1.3 Robust graph isomorphism

Our second main result concerns the ROBUSTGISO problem. Recall the definition of ROBUSTGISO from Definition 2.2.7 which we restate as follows.

Definition 7.1.3 (Re-statement of Definition 2.2.7)**.** *We say an algorithm $\mathcal{A}$ solves the* ROBUSTGISO *problem if there is a function $r : [0,1] \to [0,1]$ satisfying $r(\epsilon) \to 0$ as $\epsilon \to 0^+$ such that whenever $\mathcal{A}$ is given any pair of graphs which are $(1-\epsilon)$-isomorphic, $\mathcal{A}$ outputs a $(1-r(\epsilon))$-isomorphism between them.*[3]

[3] We could also consider the easier task of distinguishing pairs which are $(1-\epsilon)$-isomorphic from pairs

To understand the motivation of this definition, we might imagine an algorithm trying to recover an isomorphism between G and H, where H is formed by permuting the vertices of G and then introducing a small amount of *noise* — say, adding and deleting an ϵ fraction of edges. Thinking of GISO as a CSP, we are concerned with finding "almost-satisfying" solutions on "almost-satisfiable" instances. For example, suppose we are given graphs G and H which are promised to be $(1-\epsilon)$-isomorphic. Can we efficiently find a $(1-2\epsilon)$-isomorphism? A $(1-\sqrt{\epsilon})$-isomorphism? A $(1-\frac{1}{\log(1/\epsilon)})$-isomorphism? Therefore the notion of ROBUSTGISO naturally comes in.

Remark 7.1.1. *In particular, $\mathcal{A}$ in Definition 7.1.3 must solve the* GISO *problem, because given isomorphic graphs with at most m edges we can always take $\epsilon > 0$ small enough so that $r(\epsilon) < 1/m$.*

The analogous problem of robust satisfaction algorithms for CSPs over constant-size domains was introduced by Zwick [228] and has proved to be very interesting. Guruswami and Zhou [113] conjectured that the CSPs which have efficient robust algorithms (subject to $\mathsf{P} \neq \mathsf{NP}$) are precisely those of "bounded width" — roughly speaking, those that do not encode equations over abelian groups. This conjecture was recently confirmed by Barto and Kozik [38], following partial progress in [153, 72].

The graph isomorphism seems to share some of the flavor of "unbounded width" CSPs such as 3-XOR; these CSPs have the property that special algebraic methods (namely, Gaussian elimination) are available on satisfiable instances, but these methods break down once there is a small amount of noise. Indeed, the $2^{O(\sqrt{n \log n})}$-time algorithm for GISO is a somewhat peculiar mix of group theory and "local methods" (namely, Weisfeiler–Lehman). Generalizing from GISO to ROBUSTGISO seems like it might rule out applicability of group-theoretic methods, thereby making the problem much harder. Our second main theorem in a sense confirms this. Roughly speaking, it shows that ROBUSTGISO is hard assuming it is hard to distinguish random 3-XOR instances from random instances with a planted solution and slight noise:

Theorem 7.1.4. *Assume Feige's* R3XOR *Hypothesis [88]. Then there is no polynomial-time algorithm for* ROBUSTGISO. *More precisely, there exists $\epsilon_0 > 0$, such that suppose there exists $\epsilon > 0$ and a $t(n)$-time algorithm which can distinguish $(1-\epsilon)$-isomorphic n-vertex, m-edge graph pairs from pairs which are not even $(1-\epsilon_0)$-isomorphic (where $m = O(n)$). Then there is a universal constant $\Delta \in \mathbb{Z}^+$ and a $t(O(n))$-time algorithm which outputs "typical" for almost all n-variable, Δn-constraint instances of the* 3-XOR *problem, yet which never outputs "typical" on instances which are $(1-\Theta(\epsilon))$-satisfiable.*

which are not $(1 - r(\epsilon))$-isomorphic. In fact, our hardness result will hold even for this easier problem.

Here we refer to:

Feige's R3XOR **Hypothesis.** *For every fixed $\epsilon > 0$, $\Delta \in Z^+$, there is no polynomial time algorithm which on almost all* 3-XOR *instances with n variables and $m = \Delta n$ constraints outputs "typical", but which never outputs "typical" on instances which an assignment satisfying at least $(1 - \epsilon)m$ constraints.*

Remark 7.1.2. *The reader may think of the output "typical" as a* certification *that the* 3-XOR *instance has no $(1 - \epsilon)$-satisfying solution. Note that with high probability the random* 3-XOR *instance will not even have a $.51$-satisfying solution. Feige originally stated his hypothesis for the random 3SAT problem rather than the random 3XOR problem, but he showed the conjectures are equivalent. See also the work of Alekhnovich [5].*

Feige's R3XOR Hypothesis is a fairly well-believed conjecture. The variation in which the XOR constraints may involve any number of variables (not just 3) is called LPN (Learning Parities with Noise) and is believed to be hard even with any $m = \mathrm{poly}(n)$ constraints. The further variation which has linear equations modulo a large prime rather than modulo 2 is called LWE (Learning With Errors) and forms the basis for a large body of cryptography. (See [200] for more on LPN and LWE.) The fastest known algorithm for solving Feige's R3XOR problem seems to be the $2^{O(n/\log n)}$-time algorithm of Blum, Kalai, and Wasserman [44]. Thus it's plausible that ROBUSTGISO requires $2^{n^{1-o(1)}}$ time, which would make it a much more difficulty problem than GISO.

We close by mentioning some related literature on approximate graph isomorphism. The problem of finding a vertex permutation which maximizes the number of edge overlaps (or minimizes the number of edge/nonedge overlaps) was perhaps first discussed by Arora, Frieze, and Kaplan [17]. They gave an additive quasi-PTAS in the case of dense graphs ($m = \Omega(n^2)$) (as we previously discussed in Chapter 4). Arvind et al. [25] recently defined and studied several variants of the approximate graph isomorphism problem. Some of their results concern the case in which G and H have noticeably different numbers of edges and one isn't "punished" for uncovered edges in H. This kind of variant is more like approximate *subgraph* isomorphism, and is much harder. (E.g., when G is a k-clique and H is a general graph the problem is roughly equivalent to the notorious DENSEST-kSUBGRAPH problem.) The result of theirs which is most relevant to the present work involves hardness of finding approximate isomorphisms in *colored* graphs. In particular, Arvind eta al. prove the following:

Theorem 7.1.5. *([25].) There is a linear-time reduction from 2XOR (modulo 2) instances $\mathcal{I}$ to pairs of* colored *graphs G, H such that G and H are α-isomorphic if and* *only if $\mathcal{I}$ has a solution satisfying at least an α-fraction of constraints. In particular it is* NP*-hard to approximate α-isomorphism for colored graphs to a factor exceeding $\frac{11}{12}$ and* UNIQUEGAMES*-hard to approximate it to a factor exceeding* $.878$ *(by results of [219, 116], [141, 174] respectively).*

In particular, the theorem holds for colored graphs in which each color class contains at most 4 vertices. However, we do not see any way of eliminating the colors and getting the analogous inapproximability results for the usual GISO problem without using gadgets that would destroy the constant-factor gap.

7.1.4 Robust asymmetry of random graphs

One of our main technical contributions is showing that random graphs are "robustly asymmetric". In doing so, we generalize the concept of an *asymmetric graph*, which is a graph whose only automorphism is the trivial identity automorphism. A line of research (see, e.g., [81, 47, 172, 145]) has shown that several distributions of random graphs produce asymmetric graphs with high probability. In their well-known $\mathcal{G}(n,p)$ model, Erdős and Rényi [81] proved that for $\frac{\ln n}{n} \leq p \leq 1 - \frac{\ln n}{n}$, $\mathcal{G}(n,p)$ is asymmetric with high probability. If we instead consider a uniformly random d-regular n-vertex graph, the sequence of works [47, 172, 145] shows that we get an asymmetric graph with high probability for any $3 \leq d \leq n-4$. In this work we will work with a third variant, the $\mathcal{G}_{n,m}$ model, in which a graph is chosen uniformly at random from all simple graphs with n vertices and m edges.

Given a graph G and a permutation π over $V(G)$, we call π an α-automorphism if the application of π on G preserves at least an α fraction of the edges. A graph G is (β,γ)-asymmetric if any γ-automorphism π has more than a fraction of $(1-\beta)$ fixed points. Intuitively, when $\beta = 1/n$, $\gamma = 0$, the property is exactly the asymmetry property; when β and γ become larger, the property requires that any permutation that is far from identity is far from an automorphism for the graph. We encourage the reader to refer to Section 7.2 for the precise definitions.

In this chapter, we show the following robust asymmetric property of $\mathcal{G}_{n,m}$.

Theorem 7.1.6. *For large enough n, suppose that $m = cn$, where $10^4 \leq c \leq n/10^{10}$. Let $\beta^* = \max\{e^{-c/6}, \frac{1}{n}\}$. With probability $(1-n^{-15})$, for all β such that $\beta^* \leq \beta \leq 1$, $\mathcal{G}_{n,m}$ is $(\beta, \beta/240)$-asymmetric.*

A couple of comments are in order. First, an $\exp(-O(c))$ lower bound on β is necessary. This is because there are at least $\lfloor \exp(-O(c)) \cdot n \rfloor$ isolated vertices in $\mathcal{G}_{n,m}$ with high probability. The permutations which only permute these isolated vertices are 1-automorphisms. Therefore, with high probability, $\mathcal{G}_{n,m}$ is not $(\exp(-\omega(c)), 0)$-asymmetric. Second, it is possible to extend our theorem to the $\mathcal{G}(n,p)$ model by showing that there exists a constant $C > 0$, such that for $\frac{C}{n} < p < \frac{1}{C}$, with high probability, $\mathcal{G}(n,p)$ is $(\beta, \beta/240)$-asymmetric for all $\beta \geq \max\{\exp(-\frac{pn}{C}), \frac{1}{n}\}$. Third, when $c \geq 6 \ln n$ (or, when $p \geq \frac{C \ln n}{n}$ in the $\mathcal{G}(n,p)$ model), we can let $\beta = \frac{1}{n}$ and obtain that $\mathcal{G}_{n,m}$ ($\mathcal{G}(n,p)$, respectively) is asymmetric with high probability – a result in the flavor of [81]. Finally, we do not work hard to optimize the constants in the theorem statement; we believe a more careful analysis would bring them down to more civilized numbers, but it is still interesting to explore the limits of these constants.

Now we briefly explain our proof techniques. Let us consider the case where c is a big constant and $\beta = 1$, so that we only need to worry about the permutations without fixed points. We would like to show that, for every such permutation π,

$$\Pr_{G \sim \mathcal{G}_{n,m}}[\pi \text{ is a } \frac{1}{240}\text{-automorphism for } G] \ll \frac{1}{n!},$$

and therefore we can union bound over all such possible permutations. In order to do this, from all $\binom{n}{2}$ possible edges, we construct $\Omega(n^2)$ disjoint pairs of edges (e, e'), which we will refer to as "bins", such that $\pi(e) = e'$. We call a bin "half-full" if exactly one edge in the pair is selected in G. It is easy to see that whenever there are more than $\frac{m}{120}$ half-full bins, π cannot be a $\frac{1}{240}$-automorphism. At this point, we would like for $\Pr_G\left[\#\text{half-full bins} < \frac{m}{120}\right] \ll \frac{1}{n!}$, and this is easy to show. Unfortunately, this method does not work when $\beta = \frac{1}{2}$. To see why, let π be a permutation with $\frac{n}{2}$ fixed points. The probability that every edge in G has fixed points of π for its endpoints is roughly $2^{-2m} = 2^{-O(n)}$. Therefore we have $\Pr_G[\pi \text{ is an automorphism for } G] \geq 2^{-O(n)}$, and this is not enough for the application of union bound (since there are more than $(n/2)! = 2^{\Omega(n \log n)}$ such permutations). A possible fix to this problem is: we first show that with high probability $(1 - n^{-\omega(1)})$, for every π with $\frac{n}{2}$ fixed points, there are many edges of G with at least one end point not fixed by π; then, conditioned on this event, we show the probability that a fixed π is not a $\frac{1}{480}$-automorphism is small enough for the union bound method. The actual proof is more involved, and it is also technically challenging to work with c as large as $\Omega(n)$, and β as small as $\frac{1}{n}$.

Finally, for our application to the GRAPHISOMORPHISM problem, we need to extend Theorem 7.1.6 to hypergraphs. More details on robust asymmetry of random hypergraphs can be found in Section 7.6.

7.1.5 Organization

In Section 7.2, we introduce the notations and the SOS/Lasserre hierarchy. In Section 7.3, we describe a reduction from 3-XOR to GISO. The completeness and soundness lemmas for reduction are proved in Section 7.4 and Section 7.5 respectively. In Section 7.6, we prove robust asymmetry property for random graphs and random hypergraphs.

Proofs of the main theorems. Theorem 7.1.2 follows from Theorem 7.2.2, Lemma 7.4.2 and Lemma 7.5.1, by choosing $c = 10^6$. Theorem 7.1.4 follows from Lemma 7.4.1 and Lemma 7.5.1, by choosing $c = \max\{10^6, \Delta\}$.

7.2 Preliminaries

We will be working with undirected graphs and hypergraphs, both of which will be denoted by $G = (V, E)$. Here, an undirected edge $e \in E$ is a set of 2 vertices $\{i, j\}$ for graphs and a set of k vertices $\{i_1, i_2, \ldots, i_k\}$ for k-uniform hypergraphs. When G is an directed graph, we use (i, j) to denote a directed edge. We also use the notation $V(G)$ to denote the vertex set of G, and $E(G)$ to denote the edge set of G.

For any two undirected graphs (or hypergraphs) $G = (V(G), E(G))$ and $H = (V(H), E(H))$ with the same number of vertices, and for any bijection $\pi : V(G) \to V(H)$, let

$$GIso(G, H; \pi) = \frac{|\{e \in E(G) : \pi(e) \in E(H)\}|}{\max\{|E(G), E(H)|\}},$$

where $\pi(e)$ is the edge obtained by applying π on each vertex incident to e. Let

$$GIso(G, H) = \max_{\pi : V(G) \to V(H)} GIso(G, H; \pi).$$

We say that an edge $e \in E(G)$ is *satisfied* by π if $\pi(e) \in E(H)$. We call π an α-*isomorphism* for G and H if $GIso(G, H; \pi) \geq \alpha$, and we say G and H are α-*isomorphic* if $GIso(G, H) \geq \alpha$.

For any permutation $\pi : V(G) \to V(G)$, let

$$\mathrm{AUT}(G; \pi) = GIso(G, G; \pi).$$

We say that π is an α-*automorphism* for G if $\mathrm{AUT}(G; \pi) \geq \alpha$.

Given a permutation π over the set V, an element $i \in V$ is a *fixed point* of π if $\pi(i) = i$.

Definition 7.2.1. *A graph (possibly hypergraph) G is (β, γ)-*asymmetric *if, for any permutation π on the vertex set of G that has at most $(1-\beta)$ fraction of the vertices as fixed points, we have* $\mathrm{AUT}(G; \pi) < 1 - \gamma$.

We extend the $\mathcal{G}_{n,m}$ random graph model to hypergraphs as follows. Let $\mathcal{G}_{n,m}^{(k)}$ be the uniform distribution over all $\binom{\binom{n}{k}}{m}$ simple k-uniform hypergraphs with n vertices and m edges.

A 3-XOR instance $\mathcal{C}$ is a collection of equations $C_1, C_2, \ldots, C_m$ over the variable set $\mathcal{X}$. Each equation C_i is of the form $x_{j_1} + x_{j_2} + x_{j_3} = b$ where $x_{j_1}, x_{j_2}, x_{j_3}$ are the variables from $\mathcal{X}$, $b \in \mathbb{Z}_2$. Given an assignment $\tau : \mathcal{X} \to \mathbb{Z}_2$, let $\mathsf{val}(\mathcal{C}; \tau)$ be the fraction of equations in $\mathcal{C}$ satisfied by τ. Let $\mathsf{val}(\mathcal{C}) = \max_{\tau:\mathcal{X}\to\mathbb{Z}_2} \mathsf{val}(\mathcal{C}; \tau)$.

7.2.1 SOS/Lasserre hierarchy

One way to formulate the SOS/Lasserre hierarchy is via the pseudo-expectation view. We briefly recall the formulation as follows. More discussion about this view can be found in [34].

We consider the feasibility of a system over n variables $(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ with the following constraints: $P_i(x) = 0$ for $i = 1, 2, \ldots, m_P$ and $Q_i(x) \geq 0$ for $j = 1, 2, \ldots, m_Q$, where all the P_i, Q_i polynomials are of degree at most d. For $r \geq d$, the degree-r SOS/Lasserre hierarchy finds a pseudo-expectation operator $\widetilde{\mathbf{E}}[\cdot]$ defined on the space of real polynomials of degree at most r over intermediates $x_1, x_2, \ldots, x_n$ such that

- $\widetilde{\mathbf{E}}[1] = 1$;
- $\widetilde{\mathbf{E}}[\alpha p + \beta q] = \alpha\widetilde{\mathbf{E}}[p] + \beta\widetilde{\mathbf{E}}[q]$ for all real numbers α, β and all polynomials p and q of degree at most r;
- $\widetilde{\mathbf{E}}[p^2] \geq 0$ for all polynomials p of degree at most $r/2$;
- $\widetilde{\mathbf{E}}[P_i \cdot q] = 0$ for all $i = 1, 2, \ldots, m_P$ and all polynomials q such that $P_i \cdot q$ is of degree at most r;
- $\widetilde{\mathbf{E}}[Q_i \cdot p^2] \geq 0$ for all $i = 1, 2, \ldots, m_Q$ and all polynomials p such that $Q_i \cdot p^2$ is of degree at most r.

We call any operator $\widetilde{\mathbf{E}}[\cdot]$ a normalized linear operator if it has the first two properties listed above.

SOS/Lasserre hierarchy for 3-XOR. Let $\mathcal{C}$ be a 3-XOR instance on variable set $\mathcal{X}$. The degree-r SOS/Lasserre hierarchy for the natural integer programming for (the satisfiability of) $\mathcal{C}$ is to find a normalized linear pseudo-expectation operator $\widetilde{\mathbf{E}}[\cdot]$ defined on the space of polynomials of degree at most r over the indeterminates $(A[x \mapsto a])_{x\in\mathcal{X},a\in\mathbb{Z}_2}$ associated to $\mathcal{C}$, such that

1. $\widetilde{\mathbf{E}}[(A[x \mapsto a]^2 - A[x \mapsto a]) \cdot q] = 0$ for all $x \in \mathcal{X}$, $a \in \mathbb{Z}_2$, and polynomials q;
2. $\widetilde{\mathbf{E}}[(A[x \mapsto 0] + A[x \mapsto 1] - 1) \cdot q] = 0$ for all $x \in \mathcal{X}$ and polynomials q;
3. $\widetilde{\mathbf{E}}[(\sum_{\alpha_C \text{ satisfying } C} A[x_1 \mapsto \alpha_C(x_1)]A[x_2 \mapsto \alpha_C(x_2)]A[x_3 \mapsto \alpha_C(x_3)] - 1) \cdot q] = 0$ for each $C \in \mathcal{C}$ involving variables x_1, x_2, x_3 and all polynomials q;
4. $\widetilde{\mathbf{E}}[p^2] \geq 0$ for all polynomials p.

We say there is degree-r SOS refutation for the satisfiability of $\mathcal{C}$ if the pseudo-expectation operator with properties listed above does not exist.

SOS/Lasserre hierarchy for GIso. Let $G = (V(G), E(G))$ and $H = (V(H), E(H))$ be two graphs such that $|V(G)| = |V(E)|$, $|E(G)| = |E(H)|$. The degree-r SOS/Lasserre hierarchy for the natural integer programming the isomorphism problem between G and H is to find a normalized linear pseudo-expectation operator $\widetilde{\mathbf{E}}[\cdot]$ on the space of real polynomials of degree at most r over the indeterminates $(\Pi[u \mapsto v])_{u\in V(G),v\in V(H)}$ such that:

a $\widetilde{\mathbf{E}}[(\Pi[u \mapsto v]^2 - \Pi[u \mapsto v]) \cdot q] = 0$ for all $u \in V(G)$, $v \in V(H)$, and polynomials q;

b $\widetilde{\mathbf{E}}[(\sum_{v\in V(H)} \Pi[u \mapsto v] - 1) \cdot q] = 0$ for all $u \in V(G)$ and polynomials q; and similarly, $\widetilde{\mathbf{E}}[(\sum_{u\in V(G)} \Pi[u \mapsto v] - 1) \cdot q] = 0$ for all $v \in V(H)$ and polynomials q;

c $\widetilde{\mathbf{E}}[(\sum_{\{u,u'\}\in E(G)} \sum_{v,v':\{v,v'\}\in E(H)} \Pi[u \mapsto v]\Pi[u' \mapsto v'] - |E(G)|) \cdot p^2] \geq 0$ for all polynomials p;

d $\widetilde{\mathbf{E}}[p^2] \geq 0$ for all polynomials p.

We say there is degree-r SOS refutation for the isomorphism between G and H if the pseudo-expectation operator with properties listed above does not exist.

Remark 7.2.1. *It is equivalent to replace (c) by "*$\widetilde{\mathbf{E}}[(\sum_{v,v':\{v,v'\}\in E(H)} \Pi[u \mapsto v]\Pi[u' \mapsto v'] - 1) \cdot q] = 0$ *for all* $(u, u') \in E(G)$ *and all polynomials* q*".*

7.2.2 Random 3-XOR

A random 3-XOR instance with n variables and m equations is sampled by choosing m unordered 3-tuples of variables from all possible $\binom{n}{3}$ ones, and making each 3-tuple $(x_{j_1}, x_{j_2}, x_{j_3})$ into a 3-XOR constraint $x_{j_1} + x_{j_2} + x_{j_3} = b$ with an independent random $b \in \mathbb{Z}_2$.

Theorem 7.2.2. *[204] For every constant $c > 1$, there is exists $\eta > 0$ such that with probability $1 - o(1)$, the satisfiability of a random* 3-XOR *instance*[4] *with n variables and cn equations cannot be refuted by degree-(ηn) SOS/Lasserre hierarchy.*

7.3 Reduction from 3-XOR to GISO

We define a slight variant of the basic gadget from [55]:

Definition 7.3.1. *Let C be a* 3-XOR *constraint involving variables x_1, x_2, x_3. The associated gadget graph G_C consists of: 6 "variable vertices" with names "$x_i \mapsto a$" for each $i \in [3]$, $a \in \mathbb{Z}_2$; and, 4 "constraint vertices" with names "$x_1 \mapsto a_1, x_2 \mapsto a_2, x_3 \mapsto a_3$" for each partial assignment which satisfies the constraint C. Regarding edges, each pair of variable vertices $x_i \mapsto 0$, $x_i \mapsto 1$ is connected by an edge; the four constraint vertices are connected by a clique; and, each constraint vertex α is connected to the three variable vertices it is consistent with.*

Now we describe how an entire instance of 3-XOR is encoded by a graph:

Definition 7.3.2. *Let $\mathcal{C}$ be a collection of* 3-XOR *constraints over variable set $\mathcal{X}$. We define the associated graph $G_{\mathcal{C}}$ as follows: For each constraint $C \in \mathcal{C}$, the graph $G_{\mathcal{C}}$ contains a copy of the gadget graph G_C. However we* identify *all of the variable vertices $x \mapsto a$ across $x \in \mathcal{X}$, $a \in Z_2$ as well as the variable edges $(x \mapsto 0, x \mapsto 1)$. The constraint vertices associated to C, on the other hand, are left as-is, and will be named α_C. We denote the set of vertices $\{x \mapsto 0, x \mapsto 1\}$ by V_x for every variable x, denote the set of vertices corresponding to C by V_C for every variable C.*

Remark 7.3.1. *If $\mathcal{C}$ is a* 3-XOR *instance with n vertices and m constraints then the graph $G_{\mathcal{C}}$ has $N = 4m + 2n$ vertices and $M = 18m + n$ edges.*

Finally, we introduce the following notation:

[4] The random 3-XOR distribution used in [204] is slightly different, but the theorem still holds for our distribution.

Notation 7.3.1. *Let C be a* 3-XOR *constraint involving variables x_i, x_j, x_k. We write $\underline{C}$ for its homogeneous version, $x_i + x_j + x_k = 0$. Given a collection of* 3-XOR *constraints $\mathcal{C}$ we write $\underline{\mathcal{C}} = \{\underline{C} : C \in \mathcal{C}\}$.*

The reduction. *Given a collection of* 3-XOR *constraints $\mathcal{C}$, the corresponding* GISO *instance i $(G_{\mathcal{C}}, G_{\underline{\mathcal{C}}})$.*

7.4 Completeness

Lemma 7.4.1 (Completeness)**.** *If $\mathcal{C}$ is a* 3-XOR *instance such that $\mathsf{val}(\mathcal{C}) \geq 1 - \epsilon$, then $GIso(G_{\mathcal{C}}, G_{\underline{\mathcal{C}}}) \geq 1 - 2\epsilon/3$.*

Proof. Let τ be an assignment to the variables in $\mathcal{C}$ such that $\mathsf{val}(\mathcal{I}; \tau) \geq 1 - \epsilon$. Now we define a bijection π from the vertices in $G_{\mathcal{C}}$ to the ones in $G_{\underline{\mathcal{C}}}$ as follows.

For each variable vertex $x_j \mapsto b$, let $\pi(x_j \mapsto b) = x_j \mapsto b + \tau(x_j)$. For any equation vertex α_{C_i}, if C_i is not satisfied by τ, map it to an arbitrary vertex in $V_{\underline{C_i}}$. If C_i is satisfied by τ, let us suppose that $C_i : x_{j_1} + x_{j_2} + x_{j_3} = b$, let α' be an assignment such that $\alpha'(x_{j_t}) = \alpha(x_{j_t}) + \tau(x_{j_t})$ for all $t \in \{1, 2, 3\}$. Observe that

$$\alpha'(x_{j_1})+\alpha'(x_{j_2})+\alpha'(x_{j_3}) = (\alpha(x_{j_1})+\alpha(x_{j_2})+\alpha(x_{j_3}))+(\tau(x_{j_1})+\tau(x_{j_2})+\tau(x_{j_3})) = b+b = 0.$$

Therefore $\alpha'_{\underline{C_i}}$ is a vertex in $G_{\underline{\mathcal{C}}}$. We let π map α_{C_i} to $\alpha'_{\underline{C_i}}$.

It is straightforward to check that all the edges between equation vertices and between variable vertices are satisfied. Now we consider an edge between a equation vertex and a variable vertex, namely between α_{C_i} and $x_j \mapsto b$ where x_j is an variable in equation C_i and $\alpha(x_j) = b$. We show that the edge is satisfied by π whenever C_i is satisfied by τ. Let α' and b' be such that $\pi(\alpha_{C_i}) = \alpha'_{\underline{C_i}}$, $\pi(x_j \mapsto b) = x_j \mapsto b'$. Observe that

$$\alpha'(x_j) = \alpha(x_j) + \tau(x_j) = b + \tau(x_j) = b',$$

and this implies that there is an edge between α'_{C_i} and $x_j \mapsto b'$.

We see that the only edges in $G_{\mathcal{C}}$ which might not be satisfied by π are the ones between equation vertices and variable vertices where the corresponding equation vertex is not satisfied by τ. For each equation not satisfied, there are at most 12 such edges. Therefore there are at most $12\epsilon m$ edges not satisfied. We have

$$GIso(G_{\mathcal{C}}, G_{\underline{\mathcal{C}}}) \geq GIso(G_{\mathcal{C}}, G_{\underline{\mathcal{C}}}; \pi) \geq \frac{M - 12\epsilon m}{M} \geq 1 - \frac{2}{3}\epsilon. \qquad \square$$

7.4.1 SOS completeness

Lemma 7.4.2 (SOS completeness)**.** *Let $\mathcal{C}$ be a* 3-XOR *instance on variable set $\mathcal{X}$ and suppose that every SOS refutation of $\mathcal{C}$ requires degree exceeding r. Then every SOS refutation of the statement "$G_{\mathcal{C}}$ and $G_{\underline{\mathcal{C}}}$ are isomorphic" requires degree exceeding $r/3$.*

Proof. Since $\mathcal{C}$ cannot be refuted in degree r, there is a pseudo-expectation operator $\widetilde{\mathbf{E}}_{\mathcal{C}}[\cdot]$ defined on the space of real polynomials of degree at most r over the indeterminates $(A[x \mapsto a])_{x \in \mathcal{X}, a \in \mathbb{Z}_2}$ associated to $\mathcal{C}$. This $\widetilde{\mathbf{E}}_{\mathcal{C}}[\cdot]$ is normalized, linear, and satisfies:

i $\widetilde{\mathbf{E}}_{\mathcal{C}}[(A[x \mapsto a]^2 - A[x \mapsto a]) \cdot q] = 0$ for all $x \in \mathcal{X}$, $a \in \mathbb{Z}_2$, and polynomials q;

ii $\widetilde{\mathbf{E}}_{\mathcal{C}}[(A[x \mapsto 0] + A[x \mapsto 1] - 1) \cdot q] = 0$ for all $x \in \mathcal{X}$ and polynomials q;

iii $\widetilde{\mathbf{E}}_{\mathcal{C}}[(\sum_{\alpha_C \text{ satisfying } C} A[x_1 \mapsto \alpha_C(x_1)]A[x_2 \mapsto \alpha_C(x_2)]A[x_3 \mapsto \alpha_C(x_3)] - 1) \cdot q] = 0$ for each $C \in \mathcal{C}$ involving variables x_1, x_2, x_3 and all polynomials q;

iv $\widetilde{\mathbf{E}}_{\mathcal{C}}[p^2] \geq 0$ for all polynomials p.

Our task is to define a normalized linear pseudo-expectation operator $\widetilde{\mathbf{E}}_{\mathcal{G}}[\cdot]$ on the space of real polynomials of degree at most $r/3$ over the indeterminates $(\Pi[u \mapsto v])_{u \in V(G_{\mathcal{C}}), v \in V(G_{\underline{\mathcal{C}}})}$ such that:

I $\widetilde{\mathbf{E}}_{\mathcal{G}}[(\Pi[u \mapsto v]^2 - \Pi[u \mapsto v]) \cdot q] = 0$ for all $u \in V(G_{\mathcal{C}})$, $v \in V(G_{\underline{\mathcal{C}}})$, and polynomials q;

II $\widetilde{\mathbf{E}}_{\mathcal{G}}[(\sum_{v \in V(G_{\underline{\mathcal{C}}})} \Pi[u \mapsto v] - 1) \cdot q] = 0$ for all $u \in V(G_{\mathcal{C}})$ and polynomials q; and similarly, $\widetilde{\mathbf{E}}_{\mathcal{G}}[(\sum_{u \in V(G_{\mathcal{C}})} \Pi[u \mapsto v] - 1) \cdot q] = 0$ for all $v \in V(G_{\underline{\mathcal{C}}})$ and polynomials q;

III $\widetilde{\mathbf{E}}_{\mathcal{G}}[(\sum_{\{u,u'\} \in E(G_{\mathcal{C}})} \sum_{v,v' : \{v,v'\} \in E(G_{\underline{\mathcal{C}}})} \Pi[u \mapsto v]\Pi[u' \mapsto v'] - M) \cdot p^2] \geq 0$ for all polynomials p;

IV $\widetilde{\mathbf{E}}_{\mathcal{G}}[p^2] \geq 0$ for all polynomials p.

Here M denotes the number of edges in $G_{\mathcal{C}}$ (and also in $G_{\underline{\mathcal{C}}}$).

The idea is to formally define each indeterminate $\Pi[u \mapsto v]$ as a certain degree-3 multilinear polynomial of the indeterminates $A[x \mapsto a]$. Then $\widetilde{\mathbf{E}}_{\mathcal{G}}[\cdot]$ is automatically defined in terms of $\widetilde{\mathbf{E}}_{\mathcal{C}}[\cdot]$ for all polynomials of degree at most $r/3$. The natural definition is as follows:

1. Let $x \in \mathcal{X}$ and $a \in \mathbb{Z}_2$. We define $\Pi[(x \mapsto a) \mapsto (x \mapsto b)] = A[x \mapsto (a - b)]$.

2. Let $C \in \mathcal{C}$, let $\alpha_C = (x_1 \mapsto a_1, x_2 \mapsto a_2, x_3 \mapsto a_3)$ be constraint vertex in $G_{\mathcal{C}}$ corresponding to C, and let $\beta_{\underline{C}} = (x_1 \mapsto b_1, x_2 \mapsto b_2, x_3 \mapsto b_3)$ be a constraint vertex in $G_{\underline{\mathcal{C}}}$ corresponding to $\underline{C}$. We define $\Pi[\alpha_C \mapsto \beta_{\underline{C}}]$ to be the following degree-3 monomial:

$$A[x_1 \mapsto (a_1 - b_1)]A[x_2 \mapsto (a_2 - b_2)]A[x_3 \mapsto (a_3 - b_3)].$$

3. All other indeterminates $\Pi[u \mapsto v]$ are formally defined to be 0.

It is clear that $\widetilde{\mathbf{E}}_{\mathcal{G}}[\cdot]$ is normalized and linear by the same property of $\widetilde{\mathbf{E}}_{\mathcal{C}}[\cdot]$. It remains to show that the induced pseudo-expectation operator $\widetilde{\mathbf{E}}_{\mathcal{G}}[\cdot]$ satisfies (I)–(IV) using the fact that $\widetilde{\mathbf{E}}_{\mathcal{C}}[\cdot]$ satisfies (i)–(iv). Most of these are easy; for example, the implication (iv) $\Rightarrow$ (IV) is immediate. Almost as easy is that (i) $\Rightarrow$ (I) and that (i), (ii) $\Rightarrow$ (II). We illustrate some of these implications, leaving the rest to the reader. For example, let's verify (I) for indeterminates of type $\Pi[\alpha_C \mapsto \beta_{\underline{C}}]$. For brevity we'll write $\Pi[\alpha_C \mapsto \beta_{\underline{C}}]$ as $A_1A_2A_3$. Now for any polynomial q over the Π's,

$$\begin{aligned}
&\widetilde{\mathbf{E}}_{\mathcal{G}}[(\Pi[\alpha_C \mapsto \beta_{\underline{C}}]^2 - \Pi[\alpha_C \mapsto \beta_{\underline{C}}]) \cdot q] \\
&= \widetilde{\mathbf{E}}_{\mathcal{G}}[(A_1^2A_2^2A_3^2 - A_1A_2A_3) \cdot q'] && \text{(for some polynomial } q' \text{ over the } A\text{'s)} \\
&= \widetilde{\mathbf{E}}_{\mathcal{G}}[(A_1^2 - A_1)A_2^2A_3^2 \cdot q'] + \widetilde{\mathbf{E}}_{\mathcal{G}}[A_1(A_2^2 - A_2)A_3^2 \cdot q'] + \widetilde{\mathbf{E}}_{\mathcal{G}}[A_1A_2(A_3^2 - A_3) \cdot q'] \\
&= 0 && \text{(by (i).)}
\end{aligned}$$

And let's verify (II) when u is a variable vertex $x \mapsto a$:

$$\begin{aligned}
&\widetilde{\mathbf{E}}_{\mathcal{C}}\left[\left(\sum_{v \in V(G_{\underline{\mathcal{C}}})} \Pi[(x \mapsto a) \mapsto v] - 1\right) \cdot q\right] \\
&= \widetilde{\mathbf{E}}_{\mathcal{C}}[(A[x \mapsto a - 0] + A[x \mapsto a - 1] - 1) \cdot q'] && \text{(all other } \Pi\text{'s are } 0) \\
&= 0. && \text{(by (ii))}
\end{aligned}$$

The main effort is to establish (III). In fact we will show

$$\widetilde{\mathbf{E}}_{\mathcal{C}}\left[\left(\sum_{v,v' : \{v,v'\} \in E(G_{\underline{\mathcal{C}}})} \Pi[u \mapsto v]\Pi[u' \mapsto v'] - 1\right) \cdot p^2\right] = 0 \tag{7.1}$$

for all edges $\{u, u'\} \in E(G_{\mathcal{C}})$ and all p, whence (III) follows by summing. We will omit the (easy) verification of this for the edges $(x \mapsto 0, x \mapsto 1)$. Instead we will first verify that (7.1) holds for a typical clique edge associated to constraint C, say (α_C, α'_C), on variables x_1, x_2, x_3. Only the indeterminates of corresponding constraints, say $\Pi[\alpha_C \mapsto \beta_{\underline{C}}]$, are nonzero. Writing $A_i[\alpha - \beta] = A[x_i \mapsto \alpha_C(x_i) - \beta_{\underline{C}}(x_i)]$ for brevity (and similarly with primes), the quantity in (7.1) is

$$\widetilde{\mathbf{E}}_{\mathcal{C}}\Big[\Big(\sum_{\substack{\beta_{\underline{C}},\beta'_{\underline{C}} \\ \text{satisfying } \underline{C}}} A_1[\alpha-\beta]A_2[\alpha-\beta]A_3[\alpha-\beta]A_1[\alpha'-\beta']A_2[\alpha'-\beta']A_3[\alpha'-\beta'] - 1\Big) \cdot p'^2\Big]$$

$$= \widetilde{\mathbf{E}}_{\mathcal{C}}\Big[\Big(\big(\sum_{\beta_{\underline{C}}} A_1[\alpha-\beta]A_2[\alpha-\beta]A_3[\alpha-\beta]\big)\big(\sum_{\beta'_{\underline{C}}} A_1[\alpha'-\beta']A_2[\alpha'-\beta']A_3[\alpha'-\beta']\big) - 1\Big) \cdot p'^2\Big]. \tag{7.2}$$

Now for fixed α_C, as $\beta_{\underline{C}}$ ranges over all satisfying assignments to $\underline{C}$, the assignment $\alpha_C - \beta_{\underline{C}}$ ranges over all satisfying assignments to C. The analogous statement holds also for α'_C. It's now straightforward to see that the vanishing of (7.2) follows from (iii).

Our final task is to verify (7.1) also for edges between variable vertices and constraint vertices. Fix a typical such edge, say one connecting $x_1 \mapsto a_1$ to α_C. (We'll use the same notation as before for α_C; in particular, note that we must have $\alpha_C(x_1) = a_1$.) Now in this case, the quantity in (7.1) is

$$\widetilde{\mathbf{E}}_{\mathcal{C}}\Big[\Big(\sum_{\substack{b \in \mathbb{Z}_2, \\ \beta_{\underline{C}} \text{ satisfying } \underline{C}}} A[x_1 \mapsto (a_1 - b)]A_1[\alpha-\beta]A_2[\alpha-\beta]A_3[\alpha-\beta] - 1\Big) \cdot p'^2\Big]$$

$$= \widetilde{\mathbf{E}}_{\mathcal{C}}\Big[\Big(\big(\sum_{c \in \mathbb{Z}_2} A[x_1 \mapsto c]\big)\big(\sum_{\beta_{\underline{C}}} A_1[\alpha-\beta]A_2[\alpha-\beta]A_3[\alpha-\beta]\big) - 1\Big) \cdot p'^2\Big]. \tag{7.3}$$

Again, the fact that (7.3) vanishes now easily follows from (ii), (iii). $\square$

7.5 Soundness

In this section, we prove the following soundness lemma.

Lemma 7.5.1 (Soundness)**.** *Let* $\mathcal{C} = \{C_1, C_2, \dots, C_m\}$ *be a random* 3-XOR *instance with* n *variables and* $m = cn$ *equations where* $c \geq 10^10$. *With probability* $1 - o(1)$, *we have*

$$GIso(G_C, G_{\underline{C}}) < 1 - \frac{1}{95c^2}.$$

Before proving Lemma 7.5.1, we first introduce the following definition.

Definition 7.5.2. *A graph (possibly hypergraph) G is (ϵ, D)*-degree bounded *if the average degree of every set of ϵ fraction of vertices is at most D.*

Claim 7.5.3. *Suppose $c \geq 3$. A random 3-uniform hypergraph H drawn from $\mathcal{G}^{(3)}_{n,m}$, where $m = cn$, is $(1/c, 100c)$-degree bounded with probability $1 - o(1)$.*

Claim 7.5.4. *Given an (ϵ, D)-degree bounded graph G with n vertices, every set of βn vertices has at most $(\epsilon + \beta)Dn$ edges incident to them.*

Lemma 7.5.1 is directed implied by the following two lemmas.

Lemma 7.5.5. *Let $H = ([n], E = \{e_i\})$ be a 3-uniform hypergraph with n vertices and $m = cn$ hyperedges. the constraint graph of a* 3-XOR *instance with n variables and $m = cn$ equations. Suppose H is $(\epsilon, 100c)$-degree bounded and (β, γ)-asymmetric, where $\gamma \geq 200\epsilon$. Let $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ be an arbitrary* 3-XOR *with n variables and m constraints based on H. (In other words, each hyperedge e_i of H connects the indices of the 3 variables used by C_i.) If we set*

$$\delta := \delta(c, \epsilon, \beta, \gamma) = \min\left\{\frac{1}{200}, \frac{\gamma}{48}, \frac{\epsilon}{95c}\right\},$$

when $GIso(G_C, G_{\underline{C}}) \geq 1 - \delta$, we have $\mathsf{val}(\mathcal{C}) \geq .9 - 100(\epsilon + \beta)$.

Lemma 7.5.6. *If $\mathcal{C}$ is a random* 3-XOR *instance with n variables and $m \geq 10000n$ equations, then with probability $1 - o(1)$, we have $\mathsf{val}(\mathcal{C}) < .51$.*

Proof of Lemma 7.5.1. Set $\epsilon = \frac{1}{c}, \gamma = \frac{200}{c}, \beta = \frac{48000}{c}$. Combining Claim 7.5.3, Lemma 7.5.6, and Theorem 7.6.9, we know that with probability $(1 - o(1))$, all of the following hold:

1. H is $(\epsilon, 100c)$-degree bounded,
2. H is (β, γ)-asymmetric, and
3. $\mathsf{val}(\mathcal{C}) < .51$.

Given that these hold, assume for sake of contradiction that $GIso(G_{\mathcal{C}}, G_{\underline{\mathcal{C}}}) \geq 1 - \frac{1}{95c^2}$. Then because G satisfies Item 1 and Item 2, Lemma 7.5.5 implies that

$$\mathsf{val}(\mathcal{C}) \geq .9 - 100\left(\frac{1}{c} + \frac{48000}{c}\right) \geq .8,$$

where the last step follows because $c \geq 10^{10}$. However, this contradicts Item 3. Therefore, $GIso(G_{\mathcal{C}}, G_{\underline{\mathcal{C}}}) < 1 - \frac{1}{95c^2}$ with probability $1 - o(1)$. □

The proof of Lemma 7.5.6 is standard.

Proof of Lemma 7.5.6. Fix an assignment to the n variables. The probability that the assignment satisfies at least $.51m$ equations of a random 3-XOR instance is at most $\exp(-.02^2 \cdot .5m/2) = \exp(-.0001m)$ by the Chernoff bound. Since there are only 2^n assignments, the probability that no assignment satisfies more than $.5m$ equations is at least $1 - \exp(-.0001m) \cdot 2^n = 1 - o(1)$ when $m \geq 10000n$. $\square$

The rest of the section is devoted to the proof of Lemma 7.5.5.

Proof of Lemma 7.5.5. Let π be a bijection mapping the vertices in $G_{\mathcal{C}}$ to the vertices in $G_{\underline{\mathcal{C}}}$ such that $GIso(G_{\mathcal{C}}, G_{\underline{\mathcal{C}}}; \pi) \geq 1 - \delta$. We first prove that for most i's, π maps the set V_{C_i} to $V_{\underline{C_{i'}}}$ for some i', and for most j's, π maps most V_{x_j} to $V_{x'_j}$ for some j'. Formally, let A be the set of $i \in [m]$ such that $\pi(V_{C_i}) = V_{\underline{C_{i'}}}$ for some i', and let B be the set of $j \in [n]$ such that $\pi(V_{x_j}) = V_{x_{j'}}$ for some j'. We show that

Claim 7.5.7. *$|A| \geq (1 - 19\delta)m$, $|B| \geq (1 - 95c\delta)n$.*

Now we are able to define a permutation σ on the variables in $\mathcal{C}$ (as well as $\underline{\mathcal{C}}$ since the set of variables is shared). We let σ to be an arbitrary permutation so that for each $j \in B$, we have $\sigma(j) = j'$ where $\pi(V_{x_j}) = V_{x_{j'}}$. Now we show that σ is an almost automorphism for the constraint graph (i.e. the hypergraph $H = ([n], E)$).

Claim 7.5.8. *$\mathrm{AUT}(H; \sigma) \geq (1 - 100\epsilon - 24\delta)m$.*

By our setting of δ, we have $24\delta < \gamma/2$. Since we also assume that $100\epsilon < \gamma/2$, we have $\mathrm{AUT}(G; \sigma) \geq (1 - \gamma)m$, and therefore we know that σ has at least $(1 - \beta)n$ fixed points.

Now we are ready to define an assignment $\tau : \{x_j\} \to \mathbb{Z}_2$ which certifies that $\mathsf{val}(\mathcal{I}) \geq .9$. For each j which is not a fixed point of σ, define $\tau(x_j)$ arbitrarily. For each j being a fixed point of σ, we know that $\pi(V_{x_j}) = V_{x_j}$. We let $\tau(x_j) = b$ where $\pi(x_j \mapsto 0) = x_j \mapsto b$. We conclude the proof by showing the following claim.

Claim 7.5.9. *$\mathsf{val}(\mathcal{C}; \tau) \geq .9 - 100(\epsilon + \beta)$.*

$\square$

7.5.1 Proof of the claims

Proof of Claim 7.5.7. Observe that the only 4-cliques in $G_{\underline{\mathcal{C}}}$ are $V_{\underline{C_{i'}}}$ ($i' \in [m]$). Therefore, if $\pi(V_{C_i}) \neq V_{\underline{C_{i'}}}$ for every i', we know that at least one of the edges in the clique V_{C_i} is not satisfied. Therefore we have $m - |A| \leq \delta M$ (recall that $M = 18m + n$ is the number of edges in $G_{\mathcal{C}}$), i.e. $|A| \geq m - \delta M \geq (1 - 19\delta)m$.

We now know that at least $(1 - 19\delta)m \cdot 4$ equation vertices in $G_{\underline{\mathcal{C}}}$ are mapped from equation vertices in $G_{\mathcal{C}}$. Therefore there are at most $4m - (1 - 19\delta)m \cdot 4 = 76\delta m$ equation vertices in $G_{\underline{\mathcal{C}}}$ being mapped from variable vertices in $G_{\mathcal{C}}$. In other words, π maps at least $2n - 76\delta m = (2 - 76c\delta)n$ variable vertices to variable vertices. Let B' be the set of j's such that both vertices in V_{x_j} is mapped to a variable vertex. We have $B' \supseteq B$ and $|B'| \geq (2 - 76c\delta)n - n = (1 - 76c\delta)n$. For each $j \in B' \setminus B$, we know that the edge in V_{x_j} is not satisfied. Therefore $|B' \setminus B| \leq \delta M$. Therefore, $|B| = |B'| - |B' \setminus B| \geq (1 - 76c\delta)n - \delta M \geq (1 - 95c\delta)n$. □

Proof of Claim 7.5.8. Let E' be the set of hyperedges in E whose vertices are all in B. Since G is $(\epsilon, 100c)$-degree bounded, and by our setting of parameters $95c\delta < \epsilon$, by Claim 7.5.4, we know that $|E'| \geq m - 100c\epsilon n = (1 - 100\epsilon)m$. Now let us consider the hyperedges in $E'' = E' \cap A$. (Also observe that $|E''| \geq (1 - 100\epsilon - 19\delta)m$.) We claim that most of the hyperedges in E'' are satisfied by σ. For every hyperedge $e_i = \{j_1, j_2, j_3\} \in E''$ that is not satisfied, we know that $\{\sigma(j_1), \sigma(j_2), \sigma(j_3)\} \notin T$. Since $i \in E'' \subseteq A$, let i' be the equation index such that $\pi(V_{C_i}) = V_{\underline{C_{i'}}}$. Since $\{\sigma(j_1), \sigma(j_2), \sigma(j_3)\} \notin E$, we have $e_{i'} \neq \{\sigma(j_1), \sigma(j_2), \sigma(j_3)\}$. Let us assume w.l.o.g. that $\sigma(j_1) \notin e_{i'}$. Then there is no edge between $V_{\underline{C_{i'}}}$ and $V_{x_{\sigma(j_1)}}$ in $G_{\underline{\mathcal{C}}}$. Therefore the 4 edges between V_{C_i} and $V_{x_{j_1}}$ in $G_{\mathcal{C}}$ are not satisfied.

We have proved that whenever there is an hyperedge in E'' not satisfied by σ, there are at least 4 edges in $G_{\mathcal{C}}$ not satisfied by π. Since π satisfies $(1 - \delta)M$ edges, there are at most $\delta M/4$ hyperedges in E'' not satisfied by σ. Therefore, we have

$$\mathrm{AUT}(H; \sigma) \geq |E''| - \delta M/4 \geq (1 - 100\epsilon - 19\delta - 5\delta)m = (1 - 100\epsilon - 24\delta)m. \quad \square$$

Proof of Claim 7.5.9. Let E' be the set of hyperedges in E whose vertices are all fixed points of σ. Since σ has at least $(1 - \beta)n$ fixed points, and H is $(\epsilon, 100c)$-degree bounded, by Claim 7.5.4, we know that $|E'| \geq m - (\epsilon + \beta) \cdot 100cn = (1 - 100(\epsilon + \beta))m$. Now consider any $e_i \in E'$, if all the edges incident to V_{C_i} are satisfied, let E'' contain i. Since there are at most $\delta M \leq 19\delta m$ edges not satisfied, we know that $|E''| \geq |E'| - 19\delta m \geq (1 - 100(\epsilon + \beta) - 19\delta)m$. We claim that for all $e_i \in E''$, the equation C_i is satisfied. Therefore we have $\mathsf{val}(\mathcal{I}; \tau) \geq 1 - 100(\epsilon + \beta) - 19\delta \geq .9 - 100(\epsilon + \beta)$, since $\delta < 1/200$.

Now we show that C_i is satisfied by τ when $e_i \in E''$. Using similar argument in the proof of Claim 7.5.8, one can show that when $e_i = \{j_1, j_2, j_3\} \in E''$, we have $\pi(V_{C_i}) = V_{\underline{C_i}}$. Also we have $\pi(V_{x_{j_t}}) = V_{x_{j_t}}$ for all $t \in \{1,2,3\}$, by the definition of E''. Let H be the induced subgraph $G_{\mathcal{C}}[V_{C_i} \cup (\cup_{t\in\{1,2,3\}} V x_{j_t})]$, let $\underline{J}$ be the induced subgraph $G_{\underline{\mathcal{C}}}[V_{\underline{C_i}} \cup (\cup_{t\in\{1,2,3\}} V_{x_{j_t}})]$. We use the following claim to conclude the proof.

Claim 7.5.10. *If π (after projected on the suitable vertices) is an isomorphism between J and $\underline{J}$, C_i is satisfied by τ.*

$\square$

It remains to prove Claim 7.5.10. We first claim the following property about our construction $\mathcal{L}(\cdot)$.

Claim 7.5.11. *Let $\mathcal{C}$ be a* 3-XOR *instance, let $C : x_{j_1} + x_{j_2} + x_{j_3} = b$ be an equation from $\mathcal{C}$. For any $b_1, b_2, b_3 \in \mathbb{Z}_2$, and any vertex α_C, the parity of the number of neighbors of α_C in $\{x_{j_1} \mapsto b_1, x_{j_2} \mapsto b_2, x_{j_3} \mapsto b_3\}$ is $b + b_1 + b_2 + b_3$.*

Now we are ready to prove Claim 7.5.10.

Proof of Claim 7.5.10. Suppose the equation C_i is $x_{j_1}+x_{j_2}+x_{j_3} = b$. Consider $\alpha_{C_i} \in V_{C_i}$, let $\alpha'_{\underline{C_i}} = \pi(\alpha_{C_i})$. By the construction of $G_{\mathcal{C}}$ and $G_{\underline{\mathcal{C}}}$, we know that $\alpha(x_{j_1}) + \alpha(x_{j_2}) + \alpha(x_{j_3}) = b$, and $\alpha'(x_{j_1}) + \alpha'(x_{j_2}) + \alpha'(x_{j_3}) = 0$. Now let the set $A = \{x_{j_1} \mapsto 0, x_{j_2} \mapsto 0, x_{j_3} \mapsto 0\}$. By Claim 7.5.11, we know that the parity of the number of neighbors of α_{C_i} in A is $b + 0 + 0 + 0 = b$. Therefore, by isomorphism, the parity of the number of neighbors of $\alpha'_{\underline{C_i}}$ in $\pi(A)$ is also b. On the other hand, by the definition of τ, we know that $\pi(A) = \{x_{j_1} \mapsto \tau(x_{j_1}), x_{j_2} \mapsto \tau(x_{j_2}), x_{j_3} \mapsto \tau(x_{j_3})\}$. By Claim 7.5.11 again, we know that the parity of the number of neighbors of $\alpha'_{\underline{C_i}}$ in $\pi(A)$ is $0 + \tau(x_{j_1}) + \tau(x_{j_2}) + \tau(x_{j_3})$. Therefore, we have that $\tau(x_{j_1}) + \tau(x_{j_2}) + \tau(x_{j_3}) = b$, i.e. C_i is satisfied by τ. $\square$

7.6 Random graphs are robustly asymmetric

In this section we prove Theorem 7.1.6.

We first set up some definitions. For any graph $G = (V, E)$, let π be a permutation over the vertices in V, we write $\mathrm{id}(\pi)$ as the number of fixed points in the permutations, that is, $\mathrm{id}(\pi) = |\{v \in V : \pi(v) = v\}|$. We define $\triangle(G, \pi(G)) = \{e : e \in E, \pi(e) \notin E\} \bigcup \{e : e \notin E, \pi(e) \in E\}$. Note that $\mathrm{AUT}(G; \pi) = |E| - \frac{1}{2}|\triangle(G, \pi(G))|$.

For any permutation π over the vertex set V, we define a directed graph $G_\pi = (\binom{V}{2}, E_\pi)$, and $(e_1, e_2) \in E_\pi$ if and only if $e_2 = \pi(e_1)$. Since each $e = \{u, v\} \in \binom{V}{2}$ has in-degree and out-degree exactly 1, we can divide G_π into disjoint unions of directed cycles. We call each directed cycle a bin, and the size of the bin is the number of elements in the cycle.

Fact 7.6.1. *For any size-1 bins, there are only two situations:*

- $e = \{u, v\}$ *where* u *and* v *are both fixed points of* π. *We call these bins* type-1 *size-1 bins. The number of type-1 size-1 bins is at most* $\binom{\mathrm{id}(\pi)}{2}$.
- $e = \{u, v\}$ *where* $\pi(u) = v$ *and* $\pi(v) = u$. *We call these classes* type-2 *size-1 bins. The number of type-2 size-1 bins is at most* $\frac{n-\mathrm{id}(\pi)}{2}$ *for any permutation* π.

Now let us consider $G \sim \mathcal{G}_{n,m}$ where $m = cn$. Let $Z = \binom{n}{2}$ be the number of possible edges from which we choose m edges. We first prove the following lemma.

Lemma 7.6.2. *Let* $\mathcal{A}$ *be the event that for any permutation* π *such that* $\mathrm{id}(\pi) = (1-\beta)n$, *the number of the edges in* G *that fall into the bins of size* ≥ 2 *is at least* $\beta m/60$ *and at most* $2\beta Z/10^9$. *Whenever* $\beta \geq \exp(-c/6)$, *we have*

$$\Pr_{G\sim\mathcal{G}_{n,m}}[\mathcal{A}] = 1 - n^{-\omega(1)}.$$

By a union bound over all the $\beta : \beta \geq \beta^*$ (where there are at most n of them), we get the following corollary.

Corollary 7.6.3. *Let* $\mathcal{B}$ *be the event that for every* $\beta \geq \beta^*$, *any permutation* π *such that* $\mathrm{id}(\pi) = (1-\beta)n$, *the number of the edges in* G *that fall into the bins of size* ≥ 2 *is at least* $\beta m/60$ *at at most* $2\beta Z/10^9$. *We have*

$$\Pr_{G\sim\mathcal{G}_{n,m}}[\mathcal{B}] = 1 - n^{-\omega(1)}.$$

Before we prove Lemma 7.6.2, we need the following lemma.

Lemma 7.6.4. *Let* $G \sim \mathcal{G}_{n,m}$. *Suppose that* $\beta \geq \beta^*$. *With probability* $1 - n^{-\omega(1)}$, *for any* $T \subseteq V$, $|T| = \beta n$, *the number of edges incident to* T *is at least least* $c\beta n/40$.

Now we prove this lemma.

For any vertex $v \in V$, its expected degree in $\mathcal{G}_{n,m}$ is $2c$. We would like to prove that the probability that the degree is at most $c/10$ is very low. Indeed, we claim a more general statement.

Claim 7.6.5. *Let W be a set of w possible edges from $\binom{V}{2}$, where $\lfloor (n-1)/2 \rfloor \le w \le n$,*

$$\Pr_{G=(V,E)\sim\mathcal{G}_{n,m}}[|E \cap W| \le c/10] \le \exp(-c/2).$$

Observe that when W is the set of possible edges incident to v, Claim 7.6.5 says that $\Pr[\deg(v) \le c/10] \le \exp(-c/3)$.

For each possible edge e, we define a random variable X_e as the indicator variable for the event that e is selected as an edge in G. Claim 7.6.5 would be a direct application of Chernoff bound if the X_e variables were independent. However, the following claim states that Chernoff bound still holds since the variables are negatively associated. (Please refer to e.g. [126] for the definition of negatively associated random variables.)

Claim 7.6.6. *The Z variables X_e are negatively associated.*

Proof. Since $\{X_e\}$ follows the permutation distribution over m 1's and $(Z-m)$ 0's, by Theorem 2.11 in [126], the claim holds. $\square$

Proof of Claim 7.6.5. Since $\mathbf{E}[X_e] = 2c/(n-1)$ for every e, we have $\mathbf{E}[|E \cap W|] = \sum_{e\in W} \mathbf{E}[X_e] = 2c|W|/(n-1) \ge c$. The claim follows by Chernoff bound for negatively associated variables. $\square$

Similarly we show that

Claim 7.6.7. *Let W be a set of possible edges, when $c \le n/10^{10}$,*

$$\Pr_{G=(V,E)\sim\mathcal{G}_{n,m}}[|E \cap W| \ge |W|/10^9] \le \exp(-3|W|/10^{10}).$$

Proof. Since $\mathbf{E}[X_e] = 2c/(n-1)$ for every e, we have $\mathbf{E}[|E \cap W|] = \sum_{e\in W} \mathbf{E}[X_e] = 2c|W|/(n-1) < 3|W|/10^{10}$. The claim follows by Chernoff bound for negatively associated variables. $\square$

Now we are ready to prove Lemma 7.6.4.

Proof of Lemma 7.6.4. Suppose the vertices of the random graph $G \sim \mathcal{G}_{n,m}$ are numbered from 1 to n. Let $X_i = \sum_{j=i+1}^{i+\lfloor (n-1)/2 \rfloor} X_{e=\{i,(j-1) \mod n+1\}}$. By Claim 7.6.5, we know that

$$\Pr[X_i \le c/10] \le \exp(-c/2).$$

Since the random variables $\{X_i\}$ are sums of disjoint sets of negatively associated random variables X_e's, we know that the X_i's are also negatively associated. Let U be the set of vertices i such that $X_i \le c/10$. We have $\mathbf{E}[|U|] \le n \cdot \exp(-c/2)$. Using Chernoff bound for negatively associated random variables, we have

$$\Pr\left[|U| \ge \frac{1}{2}\beta n\right] \le \exp\left(-\frac{1}{3} \cdot \left(\frac{\beta n/2}{\mathbf{E}[|U|]} - 1\right)^2 \mathbf{E}[|U|]\right)$$
$$= \exp\left(-\frac{1}{3} \cdot \left(\frac{\beta n/2}{\mathbf{E}[|U|]} - 1\right)\left(\frac{\beta n}{2} - \mathbf{E}[|U|]\right)\right).$$

Using $\beta \ge \exp(-c/6)$ and $c \ge 10$, we have $\beta n/2 - \mathbf{E}[|U|] \ge \beta n/4$. Therefore,

$$\Pr\left[|U| \ge \frac{1}{2}\beta n\right] \le \exp\left(-\frac{1}{3}\left(\frac{\beta}{2} \cdot \exp(c/2) - 1\right) \cdot \frac{\beta n}{4}\right)$$
$$\le \exp\left(-\frac{1}{3}\left(\frac{1}{2} \cdot \exp(c/3) - 1\right) \cdot \frac{\beta n}{4}\right) \le \exp\left(-\frac{1}{3} \cdot \frac{1}{4} \cdot \exp(c/6) \cdot \frac{\beta n}{4}\right)$$
$$\le \exp\left(-\frac{n}{48}\right) = n^{-\omega(1)},$$

where the second and fourth inequalities are because of $\beta \ge \exp(-c/6)$, and the third inequality is because of $\exp(c/3)/2 - 1 \ge \exp(c/6)/4$ for $c \ge 10$.

Therefore, with probability $1 - n^{-\omega(1)}$, there are at most $\beta n/2$ vertices with degree at most $c/10$ (since $\deg(i) \ge X_i$ for every vertex i). When this happens, for any $T \subseteq V$, $|T| = \beta \cdot n$, there are at least $(|T| - \beta n/2)$ vertices in T with degree at least $c/10$, the sum of degrees of vertices in T is at least $(|T| - \beta n/2) \cdot c/10 = c\beta n/20$, which means the number of edges incident to any vertex in T is at least $c\beta n/40$. $\square$

Proof of Lemma 7.6.2. By Lemma 7.6.4, we know that with probability $(1 - n^{-\omega(1)})$, the number of edges in G that is incident to T is at least $c\beta n/40$, for every $T \subseteq V$ and $|T| = \beta n$. Therefore, for any π with $\mathrm{id}(\pi) = (1-\beta)n$, let T^* be the non-fixed points of π. We have $|T^*| = \beta n$. As the number of edges in size-1 bins which are incident to T is at most $|T^*|/2 = \beta n/2$, the number of selected edges that in bins of size ≥ 2 is at least $c\beta n/40 - \beta n/2 \ge \beta m/60$, when $c \ge 100$.

We also need to show that with probability $(1 - n^{-\omega(1)})$, the number of selected edges in bins of size ≥ 2 (denote this number by random variable X) is at most $2\beta^* Z/10^9$. For every π such that $\mathrm{id}(\pi) = (1-\beta)n$, let W be the set of possible edges whose end vertices are not both fixed point of π. We have $|W| = Z - \binom{(1-\beta)n}{2}$, therefore $\beta Z \le W \le 2\beta Z$ (for large enough n) and $X \le |E \cap W|$. By Claim 7.6.7 we have $\Pr[|E \cap W| \ge |W|/10^9] \le$

$\exp(-3|W|/10^{10}) \leq \exp(-3 \cdot \beta Z/10^{10})$. Therefore $\Pr[X \geq 2\beta Z/10^9] \leq \exp(-3 \cdot \beta Z/10^{10}) \leq n^{-\omega(\beta n)}$. We conclude the proof by taking a union bound over $\binom{n}{\beta n} < n^{\beta n}$ ways of choosing non-fixed points for π. $\square$

Lemma 7.6.8. *Conditioned on event $\mathcal{B}$, for $10^4 \leq c \leq n/10^7$, $\beta_0 \geq \beta^*$, with probability $(1 - n^{-17})$, G is $(\beta_0, \beta_0/240)$-asymmetric.*

Proof. For any permutation π, we define a set of more fine-grained bins. We start with the bins we defined before, and split the bins of size ≥ 4 into bins of size 2 and at most one bin of size 3 as follows. Suppose the original bin contains $\{e_1, e_2, \ldots, e_l\}$, where $\pi(e_i) = e_{i+1}$ and $\pi(e_l) = e_1$, $l \geq 4$, the we have new bins which contains $\{e_1, e_2\}, \ldots, \{e_{l-1}, e_l\}$ if l is even, and $\{e_1, e_2\}, \ldots, \{e_{l-4}, e_{l-3}\}$, $\{e_{l-2}, e_{l-1}, e_l\}$ if l is odd.

For each bin of size 2 and size 3, if all the edges are in G, we call it a full bin; if none of them are in G, we call it an empty bin; otherwise, we call it a half-full bin. Fix a permutation π, let s_π be the number of half-full bins. For each half-full bin, it contributes at least one to $\triangle(G, \pi(G))$, therefore $s_\pi \leq |\triangle(G, \pi(G))| = 2(m - \mathrm{AUT}(G, \pi))$. We have

$$\begin{aligned}
&\Pr[G \text{ is } (\beta_0, \beta_0/240)\text{-asymmetric}|\mathcal{B}] \\
&\qquad = \Pr[\forall \pi : \mathrm{id}(\pi) \leq (1 - \beta_0)n, \mathrm{AUT}(G, \pi) \leq (1 - \beta_0/240)m|\mathcal{B}] \\
&\qquad \geq \Pr[\forall \pi : \mathrm{id}(\pi) \leq (1 - \beta_0)n, s_\pi \geq \beta_0 m/120|\mathcal{B}].
\end{aligned}$$

Now we turn to show that

$$\Pr[\forall \pi : \mathrm{id}(\pi) \leq (1 - \beta_0)n, s_\pi \geq \beta_0 m/120|\mathcal{B}] \geq 1 - n^{-17}.$$

To show this, we only have to prove for every β where $\beta \geq \beta_0$,

$$\Pr[\forall \pi : \mathrm{id}(\pi) = (1 - \beta)n, s_\pi \geq \beta m/120|\mathcal{B}] \geq 1 - n^{-18}, \tag{7.4}$$

and take a union bound over (at most n possible) β's.

Fix $\beta : \beta \geq \beta_0$ and fix a permutation π such that $\mathrm{id}(\pi) = (1 - \beta)n$. Let $\mathcal{C}_t$ be the event that $\mathcal{B}$ happens and there are t edges in G fall into the bins of size 2 and 3. Since $\mathcal{B}$ is a disjoint union of $\mathcal{C}_t$ for all $t : \beta m/60 \leq t \leq 2\beta Z/10^9$, we have

$$\Pr[s_\pi \leq \beta m/120|\mathcal{B}] = \sum_{t=\beta m/60}^{m} \Pr[s_\pi \leq \beta m/120|\mathcal{C}_t] \cdot \Pr[\mathcal{C}_t|\mathcal{B}].$$

We will prove that

$$\Pr[s_\pi \leq \beta m/120|\mathcal{C}_t] = \left(\binom{n}{\beta n}(\beta n)!\right)^{-1} \cdot n^{-18}, \tag{7.5}$$

and by taking a union bound over all $\binom{n}{\beta n}(\beta n)!$ possible π's, we prove (7.4).

Let $\gamma = \beta/120$. Let L be the number of possible edges in bins of size 2 and 3. We have $L = Z - \binom{(1-\beta)n}{2} - \frac{\beta n}{2} \geq \beta n^2/4$ (for large enough n). Together with $t \leq 2\beta Z/10^9$, we have $t \leq 4L/10^9 \leq L/10^8$. Let B be the number of bins of size 2 and 3. Fix t such that $\beta m/60 \leq t \leq L/10^8$. Conditioned on $\mathcal{C}_t$, the $\binom{L}{t}$ ways to select these t edges are uniformly distributed. Now we compute the number of ways such that there are at most $2\gamma m$ half-full bins. Suppose that there are i half-full bins (for $i \leq 2\gamma m \leq t/2$). There are $\binom{B}{i}$ ways to choose these bins. There are at most $\binom{B}{(t-i)/2}$ ways to choose the full bins (since $t/2 < L/2 \cdot 10^8 < B/2$). For each half-full bin, there are at most 6 ways to choose the edges in the bin. Therefore,

$$\Pr[s_\pi = i | \mathcal{C}_t] \leq 6^i \binom{B}{i}\binom{B}{\frac{t-i}{2}}\binom{L}{t}^{-1} \leq 6^i \cdot \frac{L^{\frac{t+i}{2}}}{i!\left(\frac{t-i}{2}\right)!}\left(\frac{t}{L}\right)^t = 6^i L^{-\frac{t-i}{2}} \frac{t^t}{i!\left(\frac{t-i}{2}\right)!}. \quad (7.6)$$

Since $i! \geq (i/e)^i$, we have

$$(7.6) \leq 6^i L^{-\frac{t-i}{2}} e^{\frac{t+i}{2}} \frac{t^t}{i^i\left(\frac{t-i}{2}\right)^{\frac{t-i}{2}}} \leq 6^i L^{-\frac{t-i}{2}} (2e)^{\frac{t+i}{2}} \frac{t^t}{i^i (t-i)^{\frac{t-i}{2}}}. \quad (7.7)$$

Using $i^i (t-i)^{\frac{t-i}{2}} \geq \left(\frac{t}{4}\right)^{\frac{t+i}{2}}$, we have

$$(7.7) \leq 6^i (8e)^{\frac{t+i}{2}} \left(\frac{t}{L}\right)^{\frac{t-i}{2}} \leq (48e)^{\frac{3t}{4}} \left(\frac{t}{L}\right)^{\frac{t}{4}} \leq \left(\frac{10^7 t}{L}\right)^{\frac{t}{4}}. \quad (7.8)$$

Since $\beta m/60 \leq t \leq L/10^8$, and $(10^7 t/L)^{t/4}$ is monotonically decreasing when $t < L/(10^7 e)$, we have

$$(7.8) \leq \left(\frac{10^7 \beta m}{60L}\right)^{\frac{\beta m}{240}} \leq \left(\frac{10^7 \beta cn}{60\beta n^2/4}\right)^{\frac{\beta cn}{240}} \leq \left(\frac{10^6 c}{n}\right)^{\frac{\beta cn}{240}} \leq \left(\frac{10^{10}}{n}\right)^{40\beta n} \leq \left(\frac{1}{n}\right)^{30\beta n},$$

where the second last inequality is because $10^4 \leq c \leq n/10^7$ and $(10^6 c/n)^{\beta cn/240}$ is monotonically decreasing in this range, and the last inequality is for large enough n. Observing that $\binom{n}{\beta n}(\beta n)! \leq n^{2\beta n}$ and $\beta \geq \beta^* \geq 1/n$, we proved that

$$\Pr[s_\pi = i | \mathcal{C}_t] \leq \left(\binom{n}{\beta n}(\beta n)!\right)^{-1} \cdot n^{-20}.$$

By taking a union bound over all (at most n^2 many) i's such that $i \leq 2\gamma m$, we prove (7.5). □

Theorem 7.1.6 is proved by combining Corollary 7.6.3, Lemma 7.6.8, and taking a union bound over all possible $\beta = \beta_0$ (where there are at most n of them).

7.6.1 Generalization to hypergraphs

In this subsection, we generalize Theorem 7.1.6 to random k-uniform hypergraphs for any constant $k \geq 3$.

Theorem 7.6.9. *For any constant k, there exists constant κ_k, such that for $m = cn$ where $\kappa_k \leq c \leq \binom{n}{k}/\kappa_k^3$, n large enough, if we set $\beta^* = \max\{\exp(-c/6), 1/n\}$, with probability $(1 - n^{-15})$, for all $\beta : \beta^* \leq \beta \leq 1$, a random graph H from the distribution $\mathcal{G}_{n,m}^{(k)}$ is $(\beta, \beta/240)$-asymmetric. For $k = 3$, $\kappa_3 = 10^4$ suffices.*

The proof of Theorem 7.6.9 mostly follows the lines of the proof of Theorem 7.1.6. But we need some small modifications. For simplicity, we only prove the theorem for $k = 3$. For higher k, we encourage the readers to check by themselves.

Now we work with $k = 3$. For any permutation π over the vertex set V, we define a directed graph $G_\pi^{(3)} = (\binom{V}{3}, E_\pi)$, and $(e_1, e_2) \in E_\pi$ if and only if $e_2 = \pi(e_1)$. Since each $e \in \binom{V}{3}$ has in-degree and out-degree exactly 1, we can divide $G_\pi^{(3)}$ into disjoint unions of directed cycles. Similarly as in the ordinary graph case, we call each directed cycle a bin, and the size of the bin is the number of elements in the cycle. Let $Z_3 = \binom{n}{3}$.

Fact 7.6.10. *For any size-1 bins, there are only three situations:*

- *$e = \{u, v, w\}$ where u, v and w are all fixed points of π. We call these bins* type-1 *size-1 bins.The number of type-1 size-1 bins is at most $\binom{\mathrm{id}(\pi)}{3}$.*

- *$e = \{u, v, w\}$ where one of them is a fixed point of π and the other two map to each other under π. We call these classes type-2 size-1 bins. The number of type-2 size-1 bins is at most $\mathrm{id}(\pi) \cdot \frac{n-\mathrm{id}(\pi)}{2} = O(n^2)$ for any permutation π.*

- *$e = \{u, v, w\}$ where $\pi(u) = v$ and $\pi(v) = w$ and $\pi(w) = u$. We call these classes type-3 size-1 bins. The number of type-3 size-1 bins is at most $\frac{n-\mathrm{id}(\pi)}{3}$ for any permutation π.*

The following lemma is an analogue of Lemma 7.6.2.

Lemma 7.6.11. *For any fixed β where $\beta \geq \beta^*$, let $\mathcal{D}'$ be the event that for any permutation π such that $\mathrm{id}(\pi) = (1 - \beta)n$, the number of the hyperedges in H that fall into the bins of size ≥ 2 is at least $\beta m/60$ and at most $2\beta Z_3/10^9$. We have*

$$\Pr_{H \sim \mathcal{H}_{n,m}^{(3)}} [\mathcal{D}'] = 1 - n^{-\omega(1)}.$$

By a union bound over all the $\beta : \beta \geq \beta^*$ (there are at most n of them), we get the following corollary.

Corollary 7.6.12. *Let $\mathcal{D}$ be the event that for every $\beta \geq \beta^*$, any permutation π such that $\mathrm{id}(\pi) = (1-\beta)n$, the number of the hyperedges in H that fall into the bins of size ≥ 2 is at least $\beta m/60$ at at most $2\beta Z_3/10^9$. We have:*

$$\Pr_{H \sim \mathcal{H}_{n,m}^{(3)}}[\mathcal{D}] = 1 - n^{-\omega(1)}.$$

The proof of Lemma 7.6.11 is similar to that of Lemma 7.6.2, except that now we also need to take care of type-2 size-1 bins.

Lemma 7.6.13. *With probability $1 - n^{\omega(1)}$, for any permutation π with $\mathrm{id}(\pi) \leq (1-\beta^*)n$, the number of selected hyperedges in type-2 size-1 bins is at most $c\beta n/1000$.*

Proof. We first prove that for every $\beta \geq \beta^*$, with probability $1 - n^{-\omega(1)}$, for every permutation π with $\mathrm{id}(\pi) = (1-\beta)n$, the number of selected hyperedges in type-2 size-1 bins is at most $c\beta n/1000$. By a union bound over all possible β's (where there are at most n of them), we get the desired statement.

For any fixed permutation π with $\mathrm{id}(\pi) = (1-\beta)n$, the number of type-2 size-1 bins is at most $(1-\beta)n\beta n/2 \leq \beta n^2/2$. For each possible hyperedge e, we define the random variable X_e as the indicator variable for the event that e is selected as an hyperedge in H. Note that $\mathbf{E}[X_e] = cn/Z_3 \leq \frac{7c}{n^2}$. Define random variable $X = \sum_{e \text{ in type-2 size-1 bin}} X_e$ as the number of selected hyperedges in type-2 size-1 bins, by linearity of expectation,

$$\mathbf{E}[X] \leq \frac{\beta n^2}{2} \cdot \frac{7c}{n^2} < 4c\beta.$$

On the other hand, we can also show that all these random variables are negative associated, therefore through Chernoff bound for negative associated random variables, we have

$$\Pr_{H \sim \mathcal{G}_{n,m}^{(3)}}[X \geq \beta cn/1000] \leq \exp(-1/3 \cdot (n/250-1)^2 \cdot 4c \cdot \beta) \leq \exp(-c\beta n^2/10^5)$$

By a union bound over at most $\binom{n}{\beta n}(\beta n)! \leq n^{2\beta n}$ such permutations, the probability that there exists π with $\mathrm{id}(\pi) = (1-\beta)n$ such that the number of type-2 size-1 bins is more than $c\beta n$, is at most

$$\exp(-c\beta n^2/10^5) \cdot n^{2\beta n} = \exp(-c\beta n^2/(10^5) + 2\beta n \log n) \leq \exp(-c\beta n^2/(10^6)) \leq n^{-\omega(1)}.$$

$\square$

The following lemma is an analogue of Lemma 7.6.4, and the proof is almost identical.

Lemma 7.6.14. *Let $H \sim \mathcal{G}^{(3)}_{n,m}$. Suppose that $\beta \geq \beta^*$. With probability $1 - n^{-\omega(1)}$, for any $T \subseteq V$, $|T| = \beta n$, the number of hyperedges incident to T is at least least $c\beta n/40$.*

Proof of Lemma 7.6.11. We only establish the lower bound ($\beta m/60$). The proof for upper bound is almost identical to that in Lemma 7.6.2.

Let T be the set of non-fixed points of π, then $|T| = \beta n$. By Lemma 7.6.14, we know that with probability $(1 - n^{-\omega(1)})$, there are at least $c\beta n/40$ hyperedges incident to T – all these edges are either in bins of size ≥ 2, or edges in size-1 bins of type-2 or type-3. By Lemma 7.6.13, we know that with probability $1 - n^{\omega(1)}$ the number of selected hyperedges that fall into type-2 size-1 bins is at most $c\beta n/1000$. Finally we recall that there are at most $\frac{n-\mathrm{id}(\pi)}{3} = \beta n/3$ type-3 size-1 bins.

Therefore, with probability $(1 - n^{-\omega(1)}$, the number of selected hyperedges that fall into bins of size ≥ 2 is at least $c\beta n/40 - \beta n/3 - c\beta n/1000 \geq c\beta n/60$. $\square$

Finally we state the following analogue of Lemma 7.6.8 (whose proof is also almost identical).

Lemma 7.6.15. *Conditioned on event $\mathcal{D}$, for $10^4 \leq c \leq n/10^7$, $\beta \geq \beta^*$, with probability $(1 - n^{-17})$, $H \sim \mathcal{G}^{(3)}_{n,m}$ is $(\beta, \beta/240)$-asymmetric.*

The $k = 3$ case in Theorem 7.6.9 follows from Corollary 7.6.12 and Lemma 7.6.15.

Part II

A proof complexity view of the Parrilo–Lasserre hierarchy and the success of Lasserre on hard instances for weaker hierarchies

Chapter 8

Introduction and SOS preliminaries

8.1 Introduction

In a typical constraint satisfaction problem (CSP) we are given a set of variables V to be assigned values from some finite domain Ω (often $\{0,1\}$); we are also given a set of local constraints specifying how various small groups of variables should be assigned. The task is to find an assignment to the variables which minimizes the number of unsatisfied constraints. Sometimes there may also be inviolable global constraints; for example, that no domain element is assigned to too many variables. A canonical example is the $1/3$ vs. $2/3$ BALANCEDSEPARATOR problem as we defined earlier in this thesis: given is a graph (V, E) with n vertices which must be partitioned into two "balanced" parts, each of cardinality at least $n/3$; the goal is to minimize the number of edges crossing the cut. Thoughout this chapter, we will also call this problem BALANCEDSEPARATOR.

For such problems, certifying that there is a good solution is in NP; for example, given a graph we can efficiently prove that it has a balanced cut of size at most α simply by exhibiting the cut. But what about the opposite problem, certifying that every balanced cut has size at least β? Since this problem is coNP-complete it is unlikely that there are efficient certifications for every instance; however there may be efficient certifications for specific instances or classes of instances. For example, if we consider a linear programming relaxation of a given BALANCEDSEPARATOR instance and then exhibit a dual solution of value β, this constitutes a proof that every balanced cut in the instance has size at least β.

The question is also interesting for problems in P, especially when the complexity of the proof system is taken into account. For example, given an unsatisfiable instance

$Ax = b$ of the 3Lin2 CSP (meaning the equations are over $\mathbb{F}_2$ and each involves at most 3 variables), there is always an easy-to-verify proof of unsatisfiability: a vector y such that $y^\top A = 0$ but $y^\top b \neq 0$. However finding such a proof requires a rather specialized algorithm, Gaussian Elimination. By contrast, unsatisfiable instances of the 2Lin2 CSP have simple proofs of unsatisfiability (an unsatisfiable "cycle" of variables) which can be found by a very generic "local consistency" algorithm. Indeed, one can view this algorithm as searching for all constant-width Resolution proofs of unsatisfiability; the same algorithm works for any "bounded-width CSP" [26].

Positivstellensatz proofs. In this chapter we consider a certain strong proof system for CSPs. It belongs to the well-studied class of *algebraic proof systems*, in which local constraints are represented by polynomial equations. To handle global constraints we also allow for polynomial *inequalities*; this is also natural in the context of the linear programs and semidefinite programs used by optimization algorithms. To give an example, suppose we have a BALANCEDSEPARATOR instance (V, E) with $V = [n]$. We introduce a real variable X_i for each $i \in V$. Now to say that the optimum value of the instance is larger than β is precisely equivalent to saying the following system of polynomial equations and inequalities (each of degree at most 2) is *infeasible*:

$$A = \left\{X_i^2 = X_i \;\forall i \in [n]\right\} \cup \left\{\sum_{i=1}^n X_i \geq n/3,\; \sum_{i=1}^n X_i \leq 2n/3\right\} \cup \left\{\sum_{(i,j)\in E} (X_i - X_j)^2 \leq \beta\right\}.$$

Here the first set of equations enforces $X_i \in \{0, 1\}$, encoding a cut. The second set of inequalities enforces that the cut is balanced. The final inequality states that at most β edges cross the cut. Now what would constitute a proof that A has no real solutions; i.e., that the BALANCEDSEPARATOR value exceeds β? One certificate would be a formal identity in the polynomial ring $\mathbb{R}[X_1, \ldots, X_n]$ of the following form:

$$-1 = \sum_{i=1}^n P_i \cdot (X_i^2 - X_i) + U \cdot (\sum_{i=1}^n X_i - n/3) + U' \cdot (2n/3 - \sum_{i=1}^n X_i) + V \cdot (\beta - \sum_{(i,j)\in E} (X_i - X_j)^2) + W, \tag{8.1}$$

where $P_1, \ldots, P_n \in \mathbb{R}[X_1, \ldots, X_n]$ and where $U, U', V, W \in \mathbb{R}[X_1, \ldots, X_n]$ are each *sums of squares (SOS)*, meaning of the form $Q_1^2 + Q_2^2 + \cdots + Q_m^2$ for some $Q_1, \ldots, Q_m \in \mathbb{R}[X_1, \ldots, X_n]$. Such an identity would indeed imply that A is infeasible, since substituting any solution of A into (8.1) would give a nonnegative right-hand side.

In fact, a certain refinement [187] of the *Positivstellensatz* of Krivine [151] and Stengle [215] guarantees that if A is infeasible then there is *always* a proof of the form (8.1). A generic "SOS proof system" based on the Positivstellensatz was introduced around 1999 by Grigoriev and Vorobjov [101]. As with most algebraic proof systems it can be difficult to place an a priori upper bound on the *degree* of the polynomials needed for a proof; if we insist on a fixed degree bound d then the proof system becomes incomplete. On the other hand this incomplete system has the advantage of being efficiently *automatizable*, meaning that if a proof exists it can be found in time $\mathrm{poly}(n^d)$. The algorithm uses semidefinite programming and follows from the work of Shor [210], Nesterov [177], Lasserre [156, 157] and Parrilo [185]. See Section 8.1.1 for more details.

The power of SOS. Most of the previous relevant work focused on showing SOS-degree (equivalently, Lasserre-round) lower bounds. However, in this thesis, we bring to light the importance of SOS degree *upper* bounds for the study CSP approximability. We consider the strong integrality gap instances known for the notorious UNIQUEGAMESCSP [192, 142, 35] and will (essentially) show that degree-8 SOS proofs can certify that the instances have value close to 0. Thus the generic $\mathrm{poly}(n)$-time "level-4 Lasserre SDP" algorithm refutes their having large optimal value. This is despite the fact that the instances still have value near 1 after $\Theta(\log\log n)^{1/4}$ rounds of the rather powerful Sherali–Adams SDP hierarchy [192].

We will also further explore the relevance of SOS proof complexity to the algorithmic theory of CSP approximation. Specifically, we show that the Devanur–Khot-Saket–Vishnoi [79] instances of BALANCEDSEPARATOR can have their optimal value well-certified by a degree-4 SOS proof. We also investigate the problem of SOS proofs for the Khot–Vishnoi (KV) [144] instances of MAXCUT.

8.1.1 History

We review here some of what is known about SOS proofs and SDP hierarchies; for a much more thorough discussion we recommend the monograph by Laurent [162].

Throughout this work we write $X = (X_1, \dots, X_n)$ for a sequence of indeterminates, with the number n being clear from context. We say that the real multivariate polynomial $u \in \mathbb{R}[X]$ is *sum of squares (SOS)* if $u = s_1^2 + \cdots + s_m^2$ for some $s_1, \dots, s_m \in \mathbb{R}[X]$. Any SOS polynomial is nonnegative on all of $\mathbb{R}^n$; however, as Hilbert [117] showed in 1888 there exist nonnegative polynomials which are not SOS. The first explicit example, $X_1^2X_2^2(X_1^2 + X_2^2 - 3) + 1$, was given by Motzkin in the mid-'60s. Hilbert's 17th Problem [118] asks whether every nonnegative polynomial q is the quotient of SOS polynomials; this was solved affirmatively by Artin [23].

Artin's result also follows from the Positivstellensatz, first proved (essentially) by Krivine [151] and then later independently by Stengle [215]. Interestingly, Stengle's motivation was the duality theory of linear programming. We state a special case appearing in [45]:

Positivstellensatz. *Let A be a finite set of real multivariate polynomial equations and inequalities,*

$$A = \{p_1 = 0, p_2 = 0, \dots, p_m = 0\} \cup \{q_1 \geq 0, q_2 \geq 0, \dots, q_{m'} \geq 0\},$$

with each $p_i, q_j \in \mathbb{R}[X]$. Then A is infeasible if and only if there exist polynomials $r_1, \dots, r_m$ and SOS polynomials $(u_J)_{J \subseteq [m']}$ in $\mathbb{R}[X]$ such that

$$-1 = \sum_{i=1}^{m} r_i p_i + \sum_{J \subseteq [m']} u_J \prod_{j \in J} q_j. \tag{8.2}$$

One interesting further special case occurs when A contains only equations, not inequalities. In this case the Positivstellensatz says that $p_1, \dots, p_m$ have no common real roots if and only if the ideal they generate contains $1 + u$ for some SOS u. This special case arises whenever one wants to show that a CSP (with no global constraints) is not perfectly satisfiable. (As noted by Shor [209], one can actually reduce to this case in general by replacing $q \geq 0$ with $q - Y^2 = 0$, where Y is a new indeterminate; indeed, by further substitutions of new indeterminates one can reduce to the case where all equations are quadratic.)

Proof complexity. Extending the Nullstellensatz proof system of Beame, Impagliazzo, Krajíček, Pitassi, and Pudlák [39], Grigoriev and Vorobjov [101] proposed in 1999 the natural propositional proof system based on the Positivstellensatz. The complexity measure is degree: i.e., $\max_{i,J}\{r_i p_i, u_J \prod_{j \in J} q_j\}$ in (8.2). This is a *static* proof system, meaning that one simply exhibits the refutation (8.2).[1] Grigoriev and Vorobjov showed that refuting the single equation

$$(1 - X_0 X_1)^2 + (X_1^2 - X_2)^2 + (X_2^2 - X_3)^2 + \cdots + (X_{n-1}^2 - X_n)^2 + X_n^2 = 0$$

requires a proof of degree at least 2^{n-1}. Relying on some ideas from the work of Buss, Grigoriev, Impagliazzo, and Pitassi [54], Grigoriev showed in 1999 [97, 99] that refuting any unsatisfiable system of $\mathbb{F}_2$-linear equations requires degree at least $D/2$, where D is

[1]Grigoriev and Vorobjov also proposed a certain dynamic version of the proof system, analogous to Polynomial Calculus [68]. Indeed, [164] had earlier proposed a dynamic proof system based on Positivstellensatz. We do not discuss dynamic proof systems further in this paper.

the least width needed to give a Resolution refutation. As a consequence he showed that degree $\Omega(n)$ is necessary to prove Tseitin tautologies on n-vertex regular expander graphs and to prove that the graph K_n has no perfect matching when n is odd. Grigoriev also subsequently [98] showed that the "r-Knapsack tautology" requires a proof of degree $n+1$ for any real $r \in (\frac{n}{2} - \frac{1}{2}, \frac{n}{2} + \frac{1}{2})$; this is the infeasibility of the system

$$\{X_1^2 = X_1, \ldots, X_n^2 = X_n, X_1 + \cdots + X_n = r\},$$

for r a non-integer. For more on algebraic proof complexity with inequalities, see e.g. [100].

Optimization. We now discuss algorithmic issues. Let $u \in \mathbb{R}[x]$ be a real n-variate polynomial of degree d. A most basic optimization problem is to determine $\inf_{x \in \mathbb{R}^n} u(x)$. Roughly speaking, this is equivalent (by binary search) to the problem of deciding whether $u(x) \geq \alpha$; further, there is no loss of generality in assuming $\alpha = 0$. Unfortunately, the problem of deciding whether $u \geq 0$ is NP-hard as soon as $d \geq 4$. In 1987, Shor [210] pioneered the idea of replacing the condition $u \geq 0$ with the stronger condition that u is SOS, and noted that this can be tested in $\mathrm{poly}(n^d)$ by solving an SDP feasibility problem. (Here we ignore the issue of precision in solving SDPs; see Section 8.2 for more details.) Shor made the connection to Hilbert's 17th Problem but not to Positivstellensatz.

Beginning in 2000, Parrilo [185] and Lasserre [156, 157] independently published several works taking the idea further. Parrilo emphasized the viewpoint of Positivstellensatz as a refutation system for polynomial inequalities, while Lasserre focused significant attention on the dual SDP "problem of moments". Both proposed using $\mathrm{poly}(n^d)$-time SDPs to search for degree-d Positivstellensatz refutations, for larger and larger d.

Lasserre also proposed using certain variant forms of Positivstellensatz. For example, if one is optimizing a polynomial on a *compact* semialgebraic set K then one can use SDP optimization directly (as opposed to using binary search and feasibility testing), thanks to a version of the Positivstellensatz due to Schmüdgen [203]. Furthermore, Putinar [187] showed that if K is explicitly compact ("Archimedean") — say, one of its defining inequalities is $\sum_{i=1}^n X_i^2 \leq B$ — then the Positivstellensatz certificates (8.2) only require u_J's with $|J| \leq 1$. (Both [203, 187] contained a bug, fixed in [223].) On one hand, in practice there is rarely any harm in adding an inequality $\sum_{i=1}^n X_i^2 \leq B$ with large B; on the other hand, eliminating the u_J's with $|J| > 1$ may cause the refutation degree to increase. In any case, Lasserre focused on the polynomial optimization problem

$$\inf\{p(x) \mid x \in K\}, \quad K = \{x \in \mathbb{R}^n \mid q_1(x) \geq 0, \ldots, q_m(x) \geq 0\}, \tag{8.3}$$

and proposed a hierarchy of SDP relaxations for increasing d,

$$\inf\{L(p) \mid L : \mathbb{R}[X]_d \to \mathbb{R} \text{ is a linear map}, L(1) = 1, \text{ and } L(u), L(uq_i) \geq 0 \text{ for all SOS } u\}, \tag{8.4}$$

where $\mathbb{R}[X]_d$ denotes the ring $\mathbb{R}[X]$ restricted to polynomials of degree at most d. This is a relaxation because one can take L to be the evaluation map $p \mapsto p(x^*)$ for any optimal solution x^*. We refer to (8.4) as the *degree-d Lasserre moment SDP*; when d is even it is also known as the *level-$d/2$* (or sometimes $d/2 - 1$) *Lasserre hierarchy SDP*. The semidefinite dual of (8.4) is

$$\sup\{\beta \mid p-\beta = u_0+u_1q_1+\cdots+u_mq_m \text{ for some SOS } u_0, \ldots, u_m \text{ with } \deg(u_0), \deg(u_iq_i) \leq d\}, \tag{8.5}$$

which we refer to as the *degree-d Lasserre SOS SDP*. (One can also allow for polynomial equalities in the description of K, either by replacing them with pairs of inequalities, extending the SDP formulations as in (8.2), or by factoring out by the ideal they generate [161].) Assuming K is explicitly compact, Lasserre [157] showed that the SOS SDP's value tends to the optimal value as the degree increases. If furthermore K has a nonempty interior then there is no duality gap between (8.4) and (8.5). Generally K has *empty* interior for discrete optimization problems (e.g., if it includes the constraints $X_i^2 = X_i$); however, the duality gap issue is algorithmically irrelevant since the Ellipsoid Algorithm can't distinguish an empty interior from a small interior anyway. This issue is discussed briefly in Section 8.2.

Prior optimization results. We conclude by mentioning some known positive and negative results for the Lasserre moment SDP relaxation. Around 2001, Laurent [160] considered the Lasserre hierarchy for MAXCUT with negative edge weights allowed (i.e., the 2Lin2 CSP). She showed that degree-2 Lasserre optimally solves all instances whose underlying graph is a tree, and conversely that there are non-tree instances which degree-2 Lasserre does not solve optimally. She similarly characterized the underlying graphs which degree-4 Lasserre solves optimally: the K_5-minor-free graphs. Around 2002, Laurent [159] showed that when n is odd, the degree-$(n-1)$ moment SDP relaxation for the MAXCUT problem on K_n still has value $\frac{n^2}{4}$ (whereas the optimum value is $\frac{n^2-1}{4}$); i.e., the $\lceil\frac{n+1}{2}\rceil^{\text{th}}$ level of the Lasserre hierarchy is required to obtain the optimal solution. Around 2005, Cheung [65] considered the Knapsack problem and showed that in the optimization problem

$$\inf\{X_1 + \cdots + X_n \mid X_i^2 = X_i \,\forall i,\ X_1 + \cdots + X_n \geq r\},$$

if $r = r(n) \in (0, 1)$ is sufficiently small then the Lasserre moment SDP does not find the optimal solution (namely, 1) until the degree is "maximal", namely $2n + 2$. In 2008, Schoenbeck essentially rediscovered Grigoriev's result on $\mathbb{F}_2$-linear equations from the moment side, showing that there are n-variable 3Lin2 instance of value $\frac{1}{2} + o_n(1)$ for which the degree-$\Omega(n)$ Lasserre moment relaxation still has value 1. Building on this work, Tulsiani [220] showed degree-$\Omega(n)$ integrality gap instances matching the known NP-hardness factors for a number of CSPs. Guruswami, Sinop, and Zhou [112] showed a degree-$\Omega(n)$ integrality gap instance for the BALANCEDSEPARATOR problem with factor $\alpha > 1$, even though this level of NP-hardness is not known. They also showed a degree-$\Omega(n)$ integrality gap instance for the MAXCUT problem with factor $\frac{17}{18}$. Around 2010, Karlin, Mathieu, and Nguyen [131] showed that the degree-$2t$ Lasserre moment relaxation achieves approximation ratio $1 - \frac{1}{t}$ for the general Knapsack problem.

8.1.2 Our contributions and organization of this part

In this part, we study the power of the $O(1)$-degree SOS SDP hierarchy for several central combinatorial optimization problems.

UNIQUEGAMES. We first study the $O(1)$-degree SOS SDP hierarchy for the UNIQUEGAMES instances considered in the literature (i.e. in [192, 154, 142], obtained by composing the "quotient noisy cube" instance of [144] with the long-code alphabet reduction of [141]). In Chapter 9, we prove the following theorem.

Theorem 8.1.1 (Pre-statement of Theorem 9.0.8). *Let G be an n-variable* UNIQUEGAMES *instance with label-size q of the type considered in [192, 154, 142] obtained by composing the "quotient noisy cube" instance of [144] with the long-code alphabet reduction of [141] so that the best assignment to G's variables satisfies at most an ϵ fraction of the constraints. When ϵ is sufficiently small and n is sufficiently large, there is a degree-8 SOS refutation for the statement that the best assignment to G's variables satisfy at least $1/100$ fraction of the constraints.*

Thus just the level-4 Lasserre SDP hierarchy (essentially) solves the the UNIQUEGAMES instances.

We also investigate whether the $O(1)$-degree SOS SDP hierarchy can solve known integrality gap instances of problems that are essentially *harder* than UNIQUEGAMES. We focus on two such problems: BALANCEDSEPARATOR and MAXCUT.

BALANCEDSEPARATOR. Building on work of Khot–Vishnoi [144] and Krauthgamer–Rabani [150], Devanur, Khot, Saket, and Vishnoi (DKSV) [79] gave a family of n-vertex BALANCEDSEPARATOR instances in which the optimal balanced separator cuts an $\Omega(\frac{\log\log n}{\log n})$ fraction of the edges, but for which the SDP with triangle inequalities has value $O(\frac{1}{\log n})$. This is a factor-$\Theta(\log\log n)$ integrality gap. Raghavendra and Steurer [192] show that a factor-$(\log\log n)^{\Omega(1)}$ gap persists for these instances even for $(\log\log n)^{\Omega(1)}$ rounds of the "LH SDP hierarchy". The key to analyzing the optimum value of their instances is the KKL Theorem [129] from analysis of boolean functions. In this work we give a degree-4 SOS proof of the KKL Theorem. In turn, this is used in Chapter 10 to show the following:

Theorem 8.1.2 (Pre-statement of Theorem 10.3.3)**.** *The degree-*4 *SOS relaxation for the DKSV* BALANCEDSEPARATOR *instances has value* $\Omega(\frac{\log\log n}{\log n})$.

Thus just the level-2 Lasserre SDP hierarchy (essentially) solves the DSKV BALANCED-SEPARATOR instances.

MAXCUT. Khot and -Vishnoi [144] gave integrality gap instances for the MAXCUT problem, by composing their UNIQUEGAMES instances with the Khot–Kindler–Mossel–O'Donnell [141] MAXCUT reduction. When this reduction is executed with parameter $\rho \in (-1, 0)$, one obtains n-vertex MAXCUT instances with optimal value at most $(\arccos\rho)/\pi + o_n(1)$, but for which the SDP with triangle inequalities has value $\frac{1}{2} - \frac{1}{2}\rho - o_n(1)$. In particular, for $\rho = \rho_0 \approx -.689$, this is a factor-$.878$ integrality gap (worst possible, by the Goemans–Williamson algorithm [94]). Khot and Saket [154] subsequently showed that this gap persists even for $(\log\log\log n)^{\Omega(1)}$ rounds of the Sherali–Adams SDP hierarchy. The key to analyzing the optimum value of the KV MAXCUT instances is the Majority Is Stablest Theorem from [174]. This theorem is in turn based on an Invariance Principle for nonlinear forms of random variables, together with a Gaussian isoperimetric theorem of Borell [50]. We are able to "SOS-ize" Kindler–O'Donnell's recent new proof of the latter [147] (it essentially only needs the triangle inequality); however we do not know how to prove the former for non-polynomial functionals. Thus we currently do not know how to give an SOS proof of the Majority Is Stablest Theorem.

We turn then to a weaker version of Majority Is Stablest known as the "$\frac{2}{\pi}$ Theorem", proved in [140]. This proof relies on just the Central Limit Theorem (more precisely, the Berry–Esseen Theorem). We *are* able to give an SOS proof of the CLT Theorem, although not with a fixed constant degree bound. Rather, we are able to prove it up to an additive error of δ using an SOS proof of degree $\widetilde{O}(1/\delta^2)$. Using this, as well as the SOS analysis of the KV UNIQUEGAMES instances, we are able to show the following in Chapter 11:

Theorem 8.1.3. *There exists a universal constant $C \in \mathbb{N}^+$ such that the degree-C SOS relaxation for the KV* MAXCUT *instances (with parameter $\rho_0 \approx -.689$) is within a factor $.952$ ($> .878$) of the optimum value. For general ρ, the relaxation is within a factor of $.931$ of the optimum.*

A guide to the SOS proofs. Since even conceptually simple SOS proofs can sometimes look a little complicated, we give here a brief guide to our SOS proofs. Both of our results rely on the hypercontractive inequality for $\{-1, 1\}^n$ due to [48]. We give the SOS proofs for various hypercontractive inequalities in Section 8.4. For the simplest $(2, 4)$-hypercontractive inequality (Theorem 8.4.3), the only trick is that to evade the use of Cauchy–Schwarz in the standard proofs one needs to move to a "two-function" version of the inequality. We also need SOS proofs of a few other forms of the hypercontractive inequality. Though the notation is heavy, the proofs are essentially straightforward. On the other hand, we remark that we currently do not have an SOS proof of the $2 \to 2k$ version of the inequality with sharp constant for any integer $k > 2$.

For the UNIQUEGAMES instances, we will first use hypercontractivity to prove the basic "quotient noisy cube" instances by Khot and Vishnoi [144] does not have great solution. The bulk of the remaining technical work is in lifting the soundness proof of the KKMO gadget. On a high level this proof involves the following components: **(1)** The *invariance principle* of [174], saying that low influence functions cannot distinguish between the cube and the sphere; this allows us to argue that functions that perform well on the gadget must have an influential variable (an analog of the "Majority Is Stablest" theorem), and **(2)** the *influence decoding* procedure of [141] that maps these influential functions on each local gadget into a good global assignment for the original "quotient noisy cube" instance. The invariance principle poses a special challenge, since the proof of [174] uses so called "bump" functions which are not at all low-degree polynomials. We use a weaker invariance principle, only showing that the 4 norm of a low influence function remains the same between two probability spaces that agree on the first 2 moments. Please refer to Chapter 9 for a more detailed description.

In KKL, hypercontractivity is used to prove the "Small-Set Expansion (SSE) in the Noisy Hypercube" theorem. The usual proof of this is very short, but presents a couple of challenges for SOS proofs. One challenge is the use of Hölder's inequality with exponents $4, \frac{4}{3}$. We are able to get around the fractional powers with a couple of tricks, one which is the following: if one needs to SOS-prove, say, $p \leq \sqrt{q}$ for some nonnegative polynomial q, instead prove that $p \leq \frac{\epsilon}{2} + \frac{1}{2\epsilon} \cdot q$ for all real $\epsilon > 0$. The other challenge is that the standard proof of the SSE Theorem involves division by a polynomial quantity, something we don't see how to do with SOS proofs. Still, we manage to give a short

SOS-proof of a weaker version of the SSE Theorem which is good enough for our purposes. Finally, to obtain the BALANCEDSEPARATOR result, the last step is to SOS-prove the KKL Theorem. Even the statement of the theorem involves logarithms, which does not look SOS-friendly. We get around this with a variant of the square-root trick just mentioned.

Moving to our proof of the $\frac{2}{\pi}$ Theorem, as stated, we need an SOS-proof of the Central Limit Theorem (with error bounds). Alternately phrased, we need an Invariance Theorem for linear forms of polynomials, specifically with the absolute-value functional. Although this functional is not polynomial, we can replace the required statement with something that is: namely, when $a_1, \ldots, a_n$ are indeterminates assumed to satisfy $a_1^2 + \cdots + a_n^2 = 1$, we want to upper-bound

$$\mathop{\mathbf{E}}_{\boldsymbol{x} \sim \{-1,1\}^n}[f(\boldsymbol{x})(a_1\boldsymbol{x}_1 + \cdots + a_n\boldsymbol{x}_n)] \leq \sqrt{\frac{2}{\pi}} + e,$$

where e is an error term involving $\sum_i a_i^4$, which is small when all a_i's are small. Our SOS proof of this is somewhat technically difficult. To proceed, we upper-bound the absolute-value functional to within δ by a polynomial Q of high degree; using real approximation theory, $\widetilde{O}(1/\delta^2)$ suffices. Then we prove an Invariance Theorem for linear forms with a high-degree functional; this is feasible for linear forms (but not higher-degree ones) because of their subgaussian tails. Unlike in the usual proof of the Berry–Esseen Theorem, we need the hypercontractive inequality for high norms here.

8.2 The SOS proof system and the SDP hierarchy for optimization

In this section we give formal details of the Positivstellensatz proof system of Grigoriev–Vorobjov and the associated hierarchy of SDP algorithms due to Lasserre and Parrilo. For brevity we refer to these as "SOS proofs and hierarchies".

Definition 8.2.1. *Let $X = (X_1, \ldots, X_n)$ be indeterminates, let $q_1, \ldots, q_m, r_1, \ldots, r_{m'} \in \mathbb{R}[X]$, and let*

$$A = \{q_1 \geq 0, \ldots, q_m \geq 0\} \cup \{r_1 = 0, \ldots, r_{m'} = 0\}.$$

Given $p \in \mathbb{R}[X]$ we say that A SOS-proves $p \geq 0$ with degree k*, written*

$$A \quad \vdash_k \quad p \geq 0,$$

whenever

$$\exists v_1, \ldots, v_{m'} \text{ and SOS } u_0, u_1, \ldots, u_m \text{ such that}$$

$$p = u_0 + \sum_{i=1}^{m} u_i q_i + \sum_{j=1}^{m'} v_j r_j, \qquad \textit{with } \deg(u_0), \deg(u_i q_i), \deg(v_j r_j) \le k \,\forall i \in [m], j \in [m'].$$

(Recall we say that $w \in \mathbb{R}[X]$ is SOS if $w = s_1^2 + \cdots + s_t^2$ for some $s_i \in \mathbb{R}[X]$.)

We say that A has a degree-k SOS refutation *if*

$$A \quad \vdash_k \quad -1 \ge 0.$$

Finally, when $A = \emptyset$ we will sometimes use the shorthand

$$\vdash_k \quad p \ge 0,$$

which simply means that p is SOS and $\deg(p) \le k$.

Our notation here is suggestive of a dynamic proof system, and indeed it can be helpful to think of SOS proofs this way. For example, adding deductions is not a problem:

Fact 8.2.2. *If*

$$A \quad \vdash_k \quad p \ge 0, \qquad A' \quad \vdash_{k'} \quad p' \ge 0,$$

then

$$A \cup A' \quad \vdash_{\max(k,k')} \quad p + p' \ge 0.$$

However using transitivity or multiplying together two deductions leads to a worse degree bound when applied generically:

Fact 8.2.3. *Suppose that*

$$A \quad \vdash_k \quad q_1' \ge 0, \ \ldots, \ q_\ell' \ge 0$$

(meaning $A \vdash_k q_i' \ge 0$ for each $i \in [\ell]$). Further suppose that

$$\{q_1' \ge 0, \ldots, q_\ell' \ge 0\} \quad \vdash_{k'} \quad p \ge 0.$$

Then

$$A \quad \vdash_{k+k'} \quad p \ge 0.$$

Fact 8.2.4. *Let* $A = \{q_1 \geq 0, \ldots, q_m \geq 0\}$*,* $A' = \{q_1' \geq 0, \ldots, q_{m'}' \geq 0\}$*. If*

$$A \quad \vdash_k \quad p \geq 0, \qquad A' \quad \vdash_{k'} \quad p' \geq 0,$$

then

$$A \cup A' \cup (A \cdot A') \quad \vdash_{k+k'} \quad p \cdot p' \geq 0,$$

where $A \cdot A'$ *denotes* $\{q_i \cdot q_j' \geq 0 : i \in [m], j \in [m']\}$*.*

Notice that in the above fact we had to explicitly include product inequalities into the hypotheses. This is because in general we do not have $\{q \geq 0, q' \geq 0\} \vdash qq' \geq 0$. For example:

Proposition 8.2.5. *In* $\mathbb{R}[Y, Z]$*, for every* $k \in N$*,*

$$\{Y \geq 0, Z \geq 0\} \quad \not\vdash_k \quad YZ \geq 0.$$

Indeed, for all real $\beta \geq 0$*,*

$$\{Y \geq 0, Z \geq 0\} \quad \not\vdash_k \quad YZ \geq -\beta.$$

Proof. Suppose to the contrary that

$$YZ + \beta = u_1 + Yu_2 + Zu_3 \tag{8.6}$$

for some SOS $u_1, u_2, u_3 \in \mathbb{R}[Y, Z]$. We think of the right-hand side of (8.6) as being in $\mathbb{R}[Z][Y]$. Let k_j be the degree of Y in u_j for $j = 1, 2, 3$; note that k_1, k_3 are even and k_2 is odd. Suppose first that $\max\{k_1, k_2, k_3\} = k_1$. Then we must in fact have $k_3 = k_1$ in order to cancel the Y^{k_1} term in the RHS of (8.6). But in fact such a cancelation is impossible because the coefficient on Y^{k_1} in u_1 will be an even-degree polynomial in Z, but the coefficient on Y^{k_3} in u_3 will be an odd-degree polynomial in Z. The remaining possibility is that $k_2 > k_1, k_3$. In this case we must have $k_2 = 1$, or else the degree of Y on the RHS of (8.6) will exceed 1. Thus u_1, u_2, u_3 depend only on Z; but then (8.6) forces $u_2 = Z$, contradicting the fact that u_2 is SOS. □

For more simple examples of the weakness of SOS proofs, see [170, Chap. 2.7]. Here is another one: we cannot directly prove $Y^4 \geq 1 \Rightarrow Y^2 \geq 1$.

Proposition 8.2.6. *In* $\mathbb{R}[Y]$*, for every* $k \in N$*,*

$$Y^4 \geq 1 \quad \not\vdash_k \quad Y^2 \geq 1.$$

Proof. Suppose to the contrary that one can write

$$Y^2 - 1 = u + v(Y^4 - 1) \tag{8.7}$$

with $u, v \in \mathbb{R}[Y]$ being SOS. One cannot have $v = 0$ because $Y^2 - 1$ is not SOS (consider that $0^2 - 1$ is negative). Therefore the highest-degree term in v is of the form cY^{2j} for some real $c > 0$ and some integer j. This gives a term cY^{2j+4} on the right-hand side of (8.7) which must be canceled by u. This is impossible if $\deg(u) = 2j + 4$ because the leading coefficient on u will be positive too. So $\deg(u) > 2j + 4$, but then *its* highest-degree term remains uncanceled on the right-hand side of (8.7). $\square$

On the other hand, one *can* easily SOS-prove $Y^4 \leq 1 \Rightarrow Y^2 \leq 1$; see Fact 8.3.3. Furthermore, one can $Y^4 \geq 1 \Rightarrow Y^2 \geq 1$ by contradiction:

Proposition 8.2.7. *In $\mathbb{R}[Y]$, for any $\epsilon > 0$ we have*

$$\{Y^4 \geq 1, Y^2 \leq 1 - \epsilon\} \quad \vdash_4 \quad -1 \geq 0.$$

Proof. We leave the case of $\epsilon \geq 1$ to the reader. Otherwise, write $c = 1 - \epsilon \in (0, 1)$; then

$$-1 = \tfrac{1}{1-c^2}(c + Y^2)(c - Y^2) + \tfrac{1}{1-c^2}(Y^4 - 1)$$

and both $\frac{1}{1-c^2}(c + Y^2)$ and $\frac{1}{1-c^2}$ are SOS. $\square$

These observations reveal that when fixing the degree of SOS proofs, the SDP simplifications explored by Lasserre (see Section 8.1.1) can be damaging: it may help to multiply together constraint inequalities, and direct optimization can be worse than binary searching for refutations. Thus we propose that for optimizations problems, one should generically use the SDP hierarchy proposed by Parrilo. I.e., for

$$\inf\{p(x) \mid x \in K\}, \quad K = \{x \in \mathbb{R}^n \mid q_1(x) \geq 0, \dots, q_m(x) \geq 0\},$$

one should assume that K is "explicitly compact" (say, contains the inequality $X_1^2 + \cdots + X_n^2 \leq 2^{\mathrm{poly}(n)}$) and then use binary search to (approximately) find the largest β for which

$$\{q_{i_1} q_{i_2} \cdots q_{i_t} \geq 0 : \deg(q_{i_1} q_{i_2} \cdot \cdots \cdot q_{i_t}) \leq d\} \cup \{p \leq \beta\} \quad \vdash_d \quad -1 \geq 0. \tag{8.8}$$

This can be carried out in $\mathrm{poly}(n^d, m)$ time using the Ellipsoid Algorithm.[2]

[2]Determining (8.8) amounts to checking if a matrix of variables can be PSD while satisfying some equalities. One relaxes the equalities to two-sided inequalities with some small tolerance $\delta = 2^{-\mathrm{poly}(n)}$, allowing one to run Ellipsoid. If Ellipsoid returns a feasible solution it can be made truly PSD at the expense of adding slightly more slack in the equalities. By virtue of the compactness, this can adjusted to give a valid SOS proof of $-1 + \delta' \geq 0$.

8.3 A few simple SOS preliminaries

A well-known basic fact (following from the Fundamental Theorem of Algebra) is that every nonnegative *univariate* polynomial is SOS:

Fact 8.3.1. *Suppose $p \in \mathbb{R}[X_1]$ is a univariate real polynomial such that $p(t) \geq 0$ for all real t. Then p is SOS; i.e., $\vdash_{\deg(p)} p \geq 0$.*

The following related result is credited in [162] to Fekete and Markov–Lukács, with reference also to [170]:

Fact 8.3.2. *Suppose $p \in \mathbb{R}[Y]$ is a univariate real polynomial of degree k such that $p(t) \geq 0$ for all real $a \leq t \leq b$.*
If k is odd then

$$Y \geq a,\ b \geq Y \quad \vdash_k \quad p \geq 0.$$

If k is even then

$$(b - Y)(Y - a) \geq 0 \quad \vdash_k \quad p \geq 0.$$

We now give some additional simple SOS proofs:

Fact 8.3.3. $Y^2 \leq 1 \vdash_2 Y \leq 1, Y \geq -1$.

Proof. The first follows from $1 - Y = \frac{1}{2}(1 - Y)^2 + \frac{1}{2}(1 - Y^2)$. The second follows by replacing Y by $-Y$. □

Fact 8.3.4. $\{Y \leq 1, Y \geq -1\} \vdash_3 Y^2 \leq 1$.

Proof. $1 - Y^2 = \frac{1}{2}(1 + Y)^2(1 - Y) + \frac{1}{2}(1 - Y)^2(1 + Y)$. □

Fact 8.3.5. *If $A \vdash_k Y^2 \leq Y$ ($k \geq 2$), then $A \vdash_k Y \leq 1$.*

Proof. Since $1-Y = Y-Y^2+(1-Y)^2$ and there is a degree-k SOS proof for $Y-Y^2 \geq 0$ (assuming A), we have a degree-k SOS proof for $1 - Y \geq 0$ assuming A. □

We will need an SOS proof of the fact that $Y, Z \in \{-1, 1\} \Rightarrow \frac{Y-Z}{2} \in \{-1, 0, 1\}$:

Fact 8.3.6. $Y^2 = 1, Z^2 = 1 \vdash_3 (\frac{Y-Z}{2}) = (\frac{Y-Z}{2})^3$.

Proof. $(\frac{Y-Z}{2}) - (\frac{Y-Z}{2})^3 = (\frac{3}{8}Z - \frac{1}{8}Y)(Y^2 - 1) + (\frac{1}{8}Z - \frac{3}{8}Y)(Z^2 - 1)$. □

Fact 8.3.7. *Suppose that $A \vdash_k Y \geq -1, Y \leq 1$ and that $B \vdash_\ell Z \geq W, Z \geq -W$. Then $A \cup B \vdash_{k+\ell} Z \geq YW$.*

Proof. $Z - YW = \frac{1}{2}(Z-W)(1+Y) + \frac{1}{2}(Z+W)(1-Y)$. $\square$

Fact 8.3.8. *Suppose that $A \vdash_k Y' \geq Y$ and $B \vdash_\ell Z' \geq Z$. Further suppose $A' \vdash_{k'} Y' \geq 0$ and $B' \vdash_{\ell'} Z \geq 0$. Then $A \cup B \cup A' \cup B' \vdash_{\max\{k+\ell',k'+\ell\}} Y'Z' \geq YZ$.*

Proof. This follows from $Y'Z' - YZ = Y'(Z'-Z) + Z(Y'-Y)$. $\square$

We now move to Hölder-type inequalities.

Fact 8.3.9. $\vdash_2 YZ \leq \frac{1}{2}Y^2 + \frac{1}{2}Z^2$.

Proof. $\frac{1}{2}Y^2 + \frac{1}{2}Z^2 - YZ = \frac{1}{2}(Y-Z)^2$. $\square$

More generally, by replacing Y with $\epsilon^{1/2}Y$ and Z with $\epsilon^{-1/2}Z$, we obtain:

Fact 8.3.10. $\vdash_2 YZ \leq \frac{\epsilon}{2}Y^2 + \frac{1}{2\epsilon}Z^2$ *for any real $\epsilon > 0$.*

We would also like Young's inequality for conjugate Hölder exponents $(\frac{4}{3}, 4)$, but stating it needs a trick:

Fact 8.3.11. $\vdash_4 Y^3Z \leq \frac{3}{4}Y^4 + \frac{1}{4}Z^4$.

Proof.
$$\begin{aligned}\tfrac{3}{4}Y^4 + \tfrac{1}{4}Z^4 - Y^3Z &= (\tfrac{3}{4}Y^2 + \tfrac{1}{2}YZ + \tfrac{1}{4}Z^2)(Y-Z)^2 \\ &= (\tfrac{1}{2}Y^2 + \tfrac{1}{4}(Y+Z)^2)(Y-Z)^2 = \tfrac{1}{2}Y^2(Y-Z)^2 + \tfrac{1}{4}(Y^2-Z^2)^2.\end{aligned}$$
$\square$

By replacing Y with $\epsilon^{1/4}Y$ and Z with $\epsilon^{-3/4}Z$, we obtain:

Fact 8.3.12. $\vdash_4 Y^3Z \leq \frac{3\epsilon}{4}Y^4 + \frac{1}{4\epsilon^3}Z^4$ *for any real $\epsilon > 0$.*

Fact 8.3.13. $\{Y^2 \leq Y\} \vdash_4 YZ \leq \frac{3\epsilon}{4}Y + \frac{1}{4\epsilon^3}Z^4$ *for any real $\epsilon > 0$.*

Proof. Since we have the assumption $Y^2 \leq Y$, it suffices to prove that
$$\{Y^2 \leq Y\} \vdash_4 YZ \leq \frac{2\epsilon}{4}Y + \frac{\epsilon}{4}Y^2 + \frac{1}{4\epsilon^3}Z^4.$$
This is true because
$$\frac{3\epsilon}{4}Y + \frac{\epsilon}{4}Y^2 + \frac{1}{4\epsilon^3}Z^4 - YZ = \frac{1}{4}(\sqrt{\epsilon}Y - \frac{1}{\sqrt{\epsilon^3}}Z^2)^2 + \frac{1}{2}Y^2(\frac{1}{\sqrt{\epsilon}}Z - \sqrt{\epsilon})^2 + \frac{1}{2}(Y-Y^2)(\frac{1}{\sqrt{\epsilon}}Z - \sqrt{\epsilon})^2.$$
$\square$

Fact 8.3.14. *If* $A \vdash_k Y \geq 0$ *and* $A \vdash_k Y \leq Z$*, then* $A \vdash_{2k} Y^2 \leq Z^2$*.*

Proof. We can deduce $A \vdash_k Z \geq 0$ and therefore $A \vdash_k Z + Y \geq 0$ using Fact 8.2.2. The result now follows from Fact 8.2.4 applied to $Z^2 - Y^2 = (Z+Y)(Z-Y)$. $\square$

Fact 8.3.15. $\vdash_2 \operatorname{avg}_{i\in[n]}[X_i^2] \geq (\operatorname{avg}_{i\in[n]}[X_i])^2$.

Proof. $\operatorname{avg}_{i\in[n]}[X_i^2] - (\operatorname{avg}_{i\in[n]}[X_i])^2 = \operatorname{avg}_{i,j\in[n]}[\frac{1}{2}(X_i - X_j)^2]$. $\square$

8.4 SOS proofs of hypercontractivity

In the remainder of the work we will use some standard notions from analysis of Boolean functions; see, e.g., [181]. All of our main results will require SOS proofs of the well-known hypercontractivity theorems on $\{-1,1\}^n$, first proved by Bonami [48]. To state them, recall that any function $f : \{-1,1\}^n \to \mathbb{R}$ can be viewed as a multilinear polynomial,

$$f(x) = \sum_{S\subseteq[n]} \widehat{f}(S) \prod_{i\in S} x_i, \qquad \text{where } \widehat{f}(S) = \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[f(\boldsymbol{x}) \prod_{i\in S} \boldsymbol{x}_i]. \tag{8.9}$$

Then for $\rho \in \mathbb{R}$, the linear operator T_ρ is defined by mapping the above function to

$$\mathrm{T}_\rho f(x) = \sum_{S\subseteq[n]} \rho^{|S|} \widehat{f}(S) \prod_{i\in S} x_i.$$

Now the $p = 2$, $q \geq 2$ cases of hypercontractivity can be stated as follows:

Theorem 8.4.1. *Let* $f : \{-1,1\}^n \to \mathbb{R}$*. Then for any real* $q \geq 2$*,*

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[|\mathrm{T}_{\frac{1}{\sqrt{q-1}}} f(\boldsymbol{x})|^q] \leq \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[f(\boldsymbol{x})^2]^{q/2}.$$

Theorem 8.4.2. *Let* $f : \{-1,1\}^n \to \mathbb{R}$ *have degree at most* k*. Then for any real* $q \geq 2$*,*

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[|f(\boldsymbol{x})|^q] \leq (q-1)^{(q/2)k} \cdot \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[f(\boldsymbol{x})^2]^{q/2}.$$

Note that Theorem 8.4.2 follows immediately from Theorem 8.4.1 in case f is homogeneous of degree k. It is also known that Theorem 8.4.1 and Theorem 8.4.2 (even its homogeneous version) are "equivalent", in the sense that one can be derived from the other using various analytic tricks.

As mentioned, we would ideally like to give SOS proofs of these theorems. In order to even *state* the theorems as polynomial inequalities it is required that q be an even integer. For example, when $q = 4$ we may try to SOS-prove

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[(\mathrm{T}_{\frac{1}{\sqrt{q-1}}} f(\boldsymbol{x}))^4] \le \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[f(\boldsymbol{x})^2]^2.$$

The meaning of this is that *the* 2^n *Fourier coefficients of* f *are the indeterminates*; i.e., we work over the ring $\mathbb{R}[\widehat{f}(\emptyset), \widehat{f}(\{1\}), \dots, \widehat{f}([n])]$ and would like to show that

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[f(\boldsymbol{x})^2]^2 - \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[(\mathrm{T}_{\frac{1}{\sqrt{q-1}}} f(\boldsymbol{x}))^4]$$

is a sum of squares of polynomials over the indeterminates $\widehat{f}(S)$. Sometimes we will instead use the 2^n indeterminates "$f(x)$" for $x \in \{-1,1\}^n$ — note that this is completely equivalent because the $f(x)$'s are homogeneous linear forms in the $\widehat{f}(S)$'s and vice versa; see (8.9).

When q is an even integer it is well known that Theorem 8.4.2 has a much simpler, "almost combinatorial" proof. For example, Bonami's original paper proved the homogeneous version of Theorem 8.4.2 for even integer q using nothing more "analytic" than absolute values and Cauchy–Schwarz. (Her proof even obtains a slightly sharper constant than $(q-1)^{(q/2)k}$.) The inductive proof of Theorem 8.4.2 for $q = 4$ presented in [174] is simpler still, using only Cauchy–Schwarz. It is not hard to check that these remarks also apply to Theorem 8.4.1.

Nevertheless, it's not completely trivial to obtain SOS proofs of Theorem 8.4.1 andTheorem 8.4.2 when q is an even integer, simply because the Cauchy–Schwarz inequality, $\mathbf{E}[fg] \le \sqrt{\mathbf{E}[f^2]}\sqrt{\mathbf{E}[g^2]}$, has square-roots in it. The natural substitute is the inequality $\mathbf{E}[fg] \le \frac{1}{2}\mathbf{E}[f^2] + \frac{1}{2}\mathbf{E}[g^2]$ (see Fact 8.3.9). However fitting this into the known proof of, say, the $q = 4$ case of Theorem 8.4.2 seems to require an extra trick: moving to a "two-function" version of the statement.

Theorem 8.4.3. *(SOS proof of the two-function, $q = 4$ version of Theorem 8.4.2.)*

Let $n, k_1, k_2 \in \mathbb{N}$. For each $j = 1, 2$ and each $S \subseteq [n]$ of cardinality at most k_j, introduce an indeterminate $\widehat{f_j}(S)$. For $x \in \{-1,1\}^n$, let $f_j(x)$ denote $\sum_S \widehat{f_j}(S)\prod_{i\in S} x_i$. Then

$$\vdash_4 \quad \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[f_1(\boldsymbol{x})^2 f_2(\boldsymbol{x})^2] \le 3^{k_1+k_2} \cdot \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[f_1(\boldsymbol{x})^2] \cdot \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[f_2(\boldsymbol{x})^2].$$

However, we choose to omit the proof of Theorem 8.4.3, but give an SOS proof of the $q = 4$ case of Theorem 8.4.1 using the "two-function" idea, which will imply Theorem 8.4.3. We will need a more general statement which allows for some of the ± 1 random variables to be replaced by Gaussians; this idea is also from [174].

Theorem 8.4.4. *(SOS proof of the two-function, $q = 4$ version of Theorem 8.4.1.)*

Let $n \in \mathbb{N}$. For each $j = 1, 2$ and each $S \subseteq [n]$, introduce an indeterminate $\widehat{f}_j(S)$. For each $z = (z_1, \dots, z_n) \in \mathbb{R}^n$, let

$$f_j(z) = \sum_{S \subseteq [n]} \widehat{f}_j(S) \prod_{i \in S} z_i, \qquad \mathrm{T}_{\frac{1}{\sqrt{3}}} f_j(z) = \sum_{S \subseteq [n]} (\tfrac{1}{\sqrt{3}})^{|S|} \widehat{f}_j(S) \prod_{i \in S} z_i;$$

these are homogeneous linear polynomials in the indeterminates. Let $\boldsymbol{z} = (\boldsymbol{z}_1, \dots, \boldsymbol{z}_n)$ be a random vector in which the components $\boldsymbol{z}_i$ are independent and satisfy $\mathbf{E}[\boldsymbol{z}_i] = \mathbf{E}[\boldsymbol{z}_i^3] = 0$, $\mathbf{E}[\boldsymbol{z}_i^2] = 1$, $\mathbf{E}[\boldsymbol{z}_i^4] \leq 9$. (For example, Rademachers and standard Gaussians qualify.) Then

$$\vdash_4 \quad \mathop{\mathbf{E}}_{\boldsymbol{z}}[(\mathrm{T}_{\frac{1}{\sqrt{3}}} f_1(\boldsymbol{z}))^2 \cdot (\mathrm{T}_{\frac{1}{\sqrt{3}}} f_2(\boldsymbol{z}))^2] \leq \mathop{\mathbf{E}}_{\boldsymbol{z}}[f_1(\boldsymbol{z})^2] \cdot \mathop{\mathbf{E}}_{\boldsymbol{z}}[f_2(\boldsymbol{z})^2].$$

In particular,

$$\vdash_4 \quad \mathop{\mathbf{E}}_{\boldsymbol{z}}[(\mathrm{T}_{\frac{1}{\sqrt{3}}} f_1(\boldsymbol{z}))^4] \leq \mathop{\mathbf{E}}_{\boldsymbol{z}}[f_1(\boldsymbol{z})^2]^2.$$

Just as Theorem 8.4.2 follows immediately from Theorem 8.4.1. Theorem 8.4.3 also follows immediately from Theorem 8.4.4).

Proof of Theorem 8.4.4. The proof of the theorem is by induction on n. For $n = 0$ we need to show $\vdash_4 \widehat{f}_1(\emptyset)^2 \widehat{f}_2(\emptyset)^2 \leq \widehat{f}_1(\emptyset)^2 \widehat{f}_2(\emptyset)^2$, which is trivial. For general $n \geq 1$ and $(z_1, \dots, z_n) \in \mathbb{R}^n$ we can express $f_j(z_1, \dots, z_n) = z_n d_j(z') + e_j(z')$, where $z' \in \mathbb{R}^{n-1}$ denotes $(z_1, \dots, z_{n-1})$,

$$d_j(z') = \sum_{S \ni n} \widehat{f}_j(S) \prod_{i \in S \setminus \{n\}} z_i,$$
$$e_j(z') = \sum_{S \not\ni n} \widehat{f}_j(S) \prod_{i \in S} z_i.$$

Now

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\boldsymbol{x} \sim \{-1,1\}^n}[(\mathrm{T}_{\frac{1}{\sqrt{3}}} f_1(\boldsymbol{z}))^2 \cdot (\mathrm{T}_{\frac{1}{\sqrt{3}}} f_2(\boldsymbol{z}))^2] \\
= &\mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\left(\tfrac{1}{\sqrt{3}}\boldsymbol{z}_n \cdot \mathrm{T}_{\frac{1}{\sqrt{3}}} d_1(\boldsymbol{z}') + \mathrm{T}_{\frac{1}{\sqrt{3}}} e_1(\boldsymbol{z}')\right)^2 \left(\tfrac{1}{\sqrt{3}}\boldsymbol{z}_n \cdot \mathrm{T}_{\frac{1}{\sqrt{3}}} d_2(\boldsymbol{z}') + \mathrm{T}_{\frac{1}{\sqrt{3}}} e_2(\boldsymbol{z}')\right)^2\right]
\end{aligned}$$

$$= \mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\left(\tfrac{1}{3}\boldsymbol{z}_n^2(\mathrm{T}d_1)^2 + \tfrac{2}{\sqrt{3}}\boldsymbol{z}_n(\mathrm{T}d_1)(\mathrm{T}e_1) + (\mathrm{T}e_1)^2\right)\left(\tfrac{1}{3}\boldsymbol{z}_n^2(\mathrm{T}d_2)^2 + \tfrac{2}{\sqrt{3}}\boldsymbol{z}_n(\mathrm{T}d_2)(\mathrm{T}e_2) + (\mathrm{T}e_2)^2\right)\right],$$

where we introduced the shorthand $(\mathrm{T}d_j)$ for $\mathrm{T}_{\frac{1}{\sqrt{3}}} d_j(\boldsymbol{z}')$ (and similarly for e_j). We continue by expanding the product and using linearity of expectation, $\mathbf{E}[\boldsymbol{z}_n] = \mathbf{E}[\boldsymbol{z}_n^3] = 0$, $\mathbf{E}[\boldsymbol{z}_n^2] = 1$; thus the above equals

$$\begin{aligned}\tfrac{1}{9}\,\mathbf{E}[\boldsymbol{z}_n^4]\mathop{\mathbf{E}}_{\boldsymbol{z}'}[(\mathrm{T}d_1)^2(\mathrm{T}d_2)^2] + \tfrac{1}{3}\mathop{\mathbf{E}}_{\boldsymbol{z}'}[(\mathrm{T}d_1)^2(\mathrm{T}e_2)^2] + \tfrac{1}{3}\mathop{\mathbf{E}}_{\boldsymbol{z}'}[(\mathrm{T}d_2)^2(\mathrm{T}e_1)^2] + \mathop{\mathbf{E}}_{\boldsymbol{z}'}[(\mathrm{T}e_1)^2(\mathrm{T}e_2)^2]\\ + \tfrac{4}{3}\mathop{\mathbf{E}}_{\boldsymbol{z}'}[(\mathrm{T}d_1)(\mathrm{T}e_2)\cdot(\mathrm{T}d_2)(\mathrm{T}e_1)].\end{aligned}$$

Using Fact 8.3.9 we have

$$\vdash_4 \quad \tfrac{4}{3}\mathop{\mathbf{E}}_{\boldsymbol{z}'}[(\mathrm{T}d_1)(\mathrm{T}e_2)\cdot(\mathrm{T}d_2)(\mathrm{T}e_1)] \le \tfrac{2}{3}\mathop{\mathbf{E}}_{\boldsymbol{z}'}[(\mathrm{T}d_1)^2(\mathrm{T}e_2)^2] + \tfrac{2}{3}\mathop{\mathbf{E}}_{\boldsymbol{z}'}[(\mathrm{T}d_2)^2(\mathrm{T}e_1)^2].$$

By our assumption $\mathbf{E}[\boldsymbol{z}_n^2] \le 9$ we have $\vdash_4 \tfrac{1}{9}\,\mathbf{E}[\boldsymbol{z}_n^4]\,\mathbf{E}_{\boldsymbol{z}'}[(\mathrm{T}d_1)^2(\mathrm{T}d_2)^2] \le \mathbf{E}_{\boldsymbol{z}'}[(\mathrm{T}d_1)^2(\mathrm{T}d_2)^2]$; here we are using the fact that $\mathbf{E}_{\boldsymbol{z}'}[(\mathrm{T}d_1)^2(\mathrm{T}d_2)^2]$ is SOS.[3] Thus we have shown

$$\begin{aligned}\vdash_4 \quad & \mathop{\mathbf{E}}_{\boldsymbol{z}}[(\mathrm{T}_{\frac{1}{\sqrt{3}}} f_1(\boldsymbol{z}))^2\cdot(\mathrm{T}_{\frac{1}{\sqrt{3}}} f_2(\boldsymbol{z}))^2]\\ & \le \mathop{\mathbf{E}}_{\boldsymbol{z}'}[(\mathrm{T}d_1)^2(\mathrm{T}d_2)^2] + \mathop{\mathbf{E}}_{\boldsymbol{z}'}[(\mathrm{T}d_1)^2(\mathrm{T}e_2)^2] + \mathop{\mathbf{E}}_{\boldsymbol{z}'}[(\mathrm{T}d_2)^2(\mathrm{T}e_1)^2] + \mathop{\mathbf{E}}_{\boldsymbol{z}'}[(\mathrm{T}e_1)^2(\mathrm{T}d_e)^2].\end{aligned}$$

We use induction on each of the four terms above and deduce

$$\begin{aligned}\vdash_4 \qquad & \mathop{\mathbf{E}}_{\boldsymbol{z}}[(\mathrm{T}_{\frac{1}{\sqrt{3}}} f_1(\boldsymbol{z}))^2\cdot(\mathrm{T}_{\frac{1}{\sqrt{3}}} f_2(\boldsymbol{z}))^2]\\ \le\ & \mathop{\mathbf{E}}_{\boldsymbol{z}'}[d_1(\boldsymbol{z}')^2]\mathop{\mathbf{E}}_{\boldsymbol{z}'}[d_2(\boldsymbol{z}')^2] + \mathop{\mathbf{E}}_{\boldsymbol{z}'}[d_1(\boldsymbol{z}')^2]\mathop{\mathbf{E}}_{\boldsymbol{z}'}[e_2(\boldsymbol{z}')^2]\\ & \quad + \mathop{\mathbf{E}}_{\boldsymbol{z}'}[d_2(\boldsymbol{z}')^2]\mathop{\mathbf{E}}_{\boldsymbol{z}'}[e_1(\boldsymbol{z}')^2] + \mathop{\mathbf{E}}_{\boldsymbol{z}'}[e_1(\boldsymbol{z}')^2]\mathop{\mathbf{E}}_{\boldsymbol{z}'}[e_2(\boldsymbol{z}')^2]\\ =\ & \mathop{\mathbf{E}}_{\boldsymbol{z}'}[d_1(\boldsymbol{z}')^2 + e_1(\boldsymbol{z}')^2]\cdot\mathop{\mathbf{E}}_{\boldsymbol{z}'}[d_1(\boldsymbol{z}')^2 + e_1(\boldsymbol{z}')^2]\end{aligned}$$

But it is easily verified that $\mathbf{E}_{\boldsymbol{z}}[f_j(\boldsymbol{z})^2] = \mathbf{E}_{\boldsymbol{z}'}[d_j(\boldsymbol{z}')^2 + e_j(\boldsymbol{z}')^2]$, completing the induction. $\square$

From this we can deduce Theorem 8.4.3 with the more general class of random variables.

[3]When $\boldsymbol{z}'$ is a discrete random vector this is obvious. In the general case, note that the coefficients of the polynomial in question are finite mixed moments of $\boldsymbol{z}'$. By Carathéodory's convex hull theorem we can match any finite number of moments of $\boldsymbol{z}'$ using some discrete random vector $\boldsymbol{z}''$, thereby reducing SOS-verification to the discrete case. We will use this observation in the sequel without additional comment.

Corollary 8.4.5. *Theorem 8.4.3 also holds with the more general type of random vector $\boldsymbol{z}$ from Theorem 8.4.4 in place of $\boldsymbol{x} \sim \{-1,1\}^n$.*

Proof. Begin by defining

$$\widehat{g_j}(S) = \begin{cases} 0 & \text{if } |S| > k_j, \\ \sqrt{3}^{|S|}\widehat{f_j}(S) & \text{if } |S| \le k_j \end{cases}$$

for $j = 1, 2$, and then applying Theorem 8.4.4 to g_1, g_2. This yields

$$\vdash_4 \quad \mathop{\mathbf{E}}_{\boldsymbol{z}}[f_1(\boldsymbol{z})^2 f_2(\boldsymbol{z})^2] \le \mathop{\mathbf{E}}_{\boldsymbol{z}}[\mathrm{T}_{\sqrt{3}} f_1(\boldsymbol{z})^2] \cdot \mathop{\mathbf{E}}_{\boldsymbol{z}}[\mathrm{T}_{\sqrt{3}} f_2(\boldsymbol{z})^2].$$

By a standard computation we have

$$\mathop{\mathbf{E}}_{\boldsymbol{z}}[\mathrm{T}_{\sqrt{3}} f_j(\boldsymbol{z})^2] = \sum_{i=0}^{k_j} 3^i \cdot W_j^{=i}, \quad \text{where} \quad W_j^{=i} = \sum_{|S|=i} \widehat{f_j}(S)^2, \quad j = 1, 2.$$

We also have $\mathbf{E}_{\boldsymbol{z}}[f_j(\boldsymbol{z})^2] = \sum_{i=0}^{k_j} W_j^{=i}$. Thus to complete the proof it remains to show

$$\vdash_4 \quad \Big(\sum_{i=0}^{k_1} 3^i \cdot W_1^{=i}\Big)\Big(\sum_{i'=0}^{k_2} 3^{i'} \cdot W_2^{=i'}\Big) \le \Big(\sum_{i=0}^{k_1} 3^{k_1} \cdot W_1^{=i}\Big)\Big(\sum_{i'=0}^{k_2} 3^{k_2} \cdot W_2^{=i'}\Big).$$

But after distributing out both products, this is immediate from

$$\vdash_4 \quad 3^{i+i'} \cdot W_1^{=i} \cdot W_2^{=i'} \le 3^{k_1+k_2} \cdot W_1^{=i} \cdot W_2^{=i'}$$

for each $0 \le i \le k_1, 0 \le i' \le k_2$. $\square$

We would also like to have an SOS proof of Theorem 8.4.2 for even integers $q > 4$. We content ourselves with the following slightly weaker result, the proof of which follows easily from Corollary 8.4.5:

Theorem 8.4.6. *(SOS proof of a weakened version of the two-function, even integer q case of Theorem 8.4.6.)*

Let $n, r, k_1, k_2, \dots, k_{2^r} \in \mathbb{N}$. For each $j \in [2^r]$ and each $S \subseteq [n]$ of cardinality at most k_j, introduce an indeterminate $\widehat{f_j}(S)$. Let $f_1(z), \dots, f_{2^r}(z)$ and random vector $\boldsymbol{z}$ be as in Theorem 8.4.4. Then

$$\vdash_{2^{r+1}} \quad \mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\prod_{j=1}^{2^r} f_j(\boldsymbol{z})^2\right] \le 3^{r(k_1+\cdots+k_{2^r})} \cdot \prod_{j=1}^{2^r} \mathop{\mathbf{E}}_{\boldsymbol{z}}[f_j(\boldsymbol{z})^2].$$

Proof. The proof is by induction on r. The $r = 0$ case is trivial. For $r \geq 1$, define

$$F_1(z) = \prod_{j=1}^{2^{r-1}} f_j(z), \qquad F_2(z) = \prod_{j=2^{r-1}+1}^{2^r} f_j(z).$$

Note these are degree-2^{r-1} in the indeterminates. Further, one may express

$$F_1(z) = \sum_{\substack{T \subseteq [n] \\ |T| \leq k_1 + \cdots + k_{2^{r-1}}}} \widehat{f}(T) \prod_{i \in T} z_i,$$

where $\widehat{f}(T)$ denotes a degree-2^{r-1} polynomial in the indeterminates, and similarly for F_2. Thus we may apply Corollary 8.4.5 to F_1 and F_2 and deduce

$$\vdash_{2^{r+1}} \quad \mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\prod_{j=1}^{2^r} f_j(\boldsymbol{z})^2\right] \leq 3^{k_1 + \cdots + k_{2^r}} \cdot \mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\prod_{j=1}^{2^{r-1}} f_j(\boldsymbol{z})^2\right] \cdot \mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\prod_{j=2^{r-1}+1}^{2^r} f_j(\boldsymbol{z})^2\right]. \tag{8.10}$$

By induction we have

$$\vdash_{2^r} \quad \mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\prod_{j=1}^{2^{r-1}} f_j(\boldsymbol{z})^2\right] \leq 3^{(r-1)(k_1 + \cdots + k_{2^{r-1}})} \cdot \prod_{j=1}^{2^{r-1}} \mathop{\mathbf{E}}_{\boldsymbol{z}}[f_j(\boldsymbol{z})^2],$$

$$\vdash_{2^r} \quad \mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\prod_{j=2^{r-1}+1}^{2^r} f_j(\boldsymbol{z})^2\right] \leq 3^{(r-1)(k_{2^{r-1}+1} + \cdots + k_{2^r})} \cdot \prod_{j=2^{r-1}+1}^{2^r} \mathop{\mathbf{E}}_{\boldsymbol{z}}[f_j(\boldsymbol{z})^2],$$

and all four expressions above are SOS of degree 2^r. Combining these via Fact 8.3.8 yields

$$\vdash_{2^{r+1}} \quad \mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\prod_{j=1}^{2^{r-1}} f_j(x)^2\right] \cdot \mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\prod_{j=2^{r-1}+1}^{2^r} f_j(x)^2\right]$$
$$\leq 3^{(r-1)(k_1 + \cdots + k_{2^r})} \cdot \prod_{j=1}^{2^{r-1}} \mathop{\mathbf{E}}_{\boldsymbol{z}}[f_j(\boldsymbol{z})^2] \cdot \prod_{j=2^{r-1}+1}^{2^r} \mathop{\mathbf{E}}_{\boldsymbol{z}}[f_j(\boldsymbol{z})^2],$$

which taken together with (8.10) completes the induction. $\square$

Corollary 8.4.7. *(SOS proof of a weakened version of the even integer q case of Theorem 8.4.6.)*

Let $n, k \in \mathbb{N}$. For each $S \subseteq [n]$ of cardinality at most k, introduce an indeterminate $\widehat{f}(S)$. Let $f(z)$ and random vector $\boldsymbol{z}$ be as in Theorem 8.4.4. Then for any even integer $q \geq 2$,

$$\vdash_{2^{\lceil \log_2 q \rceil}} \quad \mathop{\mathbf{E}}_{\boldsymbol{z}}[f(\boldsymbol{z})^q] \leq \sqrt{3}^{(q\lceil \log_2 q \rceil - q)k} \cdot \mathop{\mathbf{E}}_{\boldsymbol{z}}[f_j(\boldsymbol{z})^2]^{q/2}.$$

Proof. Take $r = \lceil \log_2 q \rceil - 1$, $f_1 = \cdots = f_{q/2} = f$, $f_{q/2+1} = \cdots = f_{2^r} = 1$ in Theorem 8.4.6. $\square$

Chapter 9

Analysis of the UNIQUEGAMES instances

In this section we prove the following theorem.

Theorem 9.0.8. *Let G be an n-variable* UNIQUEGAMES *instance with label-size q of the type considered in [192, 154, 142] obtained by composing the "quotient noisy cube" instance of [144] with the long-code alphabet reduction of [141] so that the best assignment to G's variables satisfies at most an ϵ fraction of the constraints. When ϵ is sufficiently small and n is sufficiently large, there is a degree-8 SOS refutation for the statement that the best assignment to G's variables satisfy at least $1/100$ fraction of the constraints.*

The more formal version of Theorem 9.0.8 is Theorem 9.4.1 later in this section. Now we give an overview of the proof.

The proof is very technical, as it is obtained by taking the already rather technical proofs of soundness for these instances, and "lifting" each step into the SOS hierarchy, a procedure that causes additional difficulties. The high level structure of all integrality gap instances constructed in the literature was the following: start with a basic integrality gap instance of UNIQUEGAMES where the Basic SDP outputs $1-o(1)$ but the true optimum is $o(1)$, the *alphabet size* of $\mathcal{G}$ is (necessarily) $N=\omega(1)$. Then, apply an *alphabet-reduction gadget* (such as the long code, or in the recent work [35] the so called "short code") to transform $\mathcal{G}$ into an instance G with some constant alphabet size q. The soundness proof of the gadget guarantees that the true optimum of G is small, while the analysis of previous works managed to "lift" the completeness proofs, and argue that the instance G survives a number of rounds that tends to infinity as ϵ tends to zero, where $(1-\epsilon)$ is the completeness value in the gap constructions, and exact tradeoff between number of rounds and ϵ depends on the paper and hierarchy.

The fact that the basic instance $\mathcal{G}$ has small integral value can be shown by appealing to hypercontractivity of low-degree polynomials, and hence can be "lifted" to the SOS world using tools developed in Section 8.4. This part of the proof is presented in Section 9.3. The bulk of the remaining technical work is in lifting the soundness proof of the gadget. On a high level this proof involves the following components: **(1)** The *invariance principle* of [174], saying that low influence functions cannot distinguish between the cube and the sphere (related proofs presented in Section 9.1); this allows us to argue that functions that perform well on the gadget must have an influential variable (related proofs presented in Section 9.2), and **(2)** the *influence decoding* procedure of [141] that maps these influential functions on each local gadget into a good global assignment for the original instance $\mathcal{G}$ (related proofs presented in Section 9.4).

The invariance principle poses a special challenge, since the proof of [174] uses so called "bump" functions which are not at all low-degree polynomials.[1] We use a weaker invariance principle, only showing that the 4 norm of a low influence function remains the same between two probability spaces that agree on the first 2 moments. Unlike the usual invariance principle, we do not move between Bernoulli variables and Gaussian space, but rather between two different distributions on the discrete cube. It turns out that for the purposes of these UNIQUEGAMES integrality gaps, the above suffices. The lifted invariance principle is proven via a "hybrid" argument similar to the argument of [174], where hypercontractivity of low-degree polynomials again plays an important role.

The soundness analysis of [141] is obtained by replacing each local function with an average over its neighbors, and then choosing a random influential coordinate from the new local function as an assignment for the original UNIQUEGAMES instance. We follow the same approach. It turns out that by making appropriate modification to the analysis, it can be lifted to complete the proof of the theorem.

9.1 An invariance principle for the fourth moment

In this section, we will be interested in $f(x)$ that is a multilinear polynomial over the doubly-indexed set of indeterminates $\{x_{i,j}\}_{i\in[N],j\in\mathbb{N}}$ with degree at most ℓ, i.e., $f(x)$ can be written in the form of

$$f(x) = \sum_{\alpha\in\mathbb{N}^N,|\alpha|\le\ell} \hat{f}(\alpha)x_\alpha,$$

where $x_\alpha = \prod_{i=1}^N x_{i,\alpha_i}$, and $|\alpha|$, the degree of α, is defined to be $|\{i \in [N] : \alpha_i > 0\}|$.

[1] A similar, though not identical, challenge arises in [35] where they need to extend the invariance principle to the "short code" setting. However, their solution does not seem to apply in our case, and we use a different approach.

We will use the following definition introduced in [174].

Definition 9.1.1 (Definition 3.1 from [174])**.** *We call a collection of finitely many orthonormal random variables, one of which is the constant* 1*, an* orthonormal ensemble*. We will write a typical* sequence *of n orthonormal ensembles as $\boldsymbol{\mathcal{X}} = (\boldsymbol{\mathcal{X}}_1, \boldsymbol{\mathcal{X}}_2, \ldots, \boldsymbol{\mathcal{X}}_N)$, where $\boldsymbol{\mathcal{X}}_i = \{\boldsymbol{X}_{i,0} = 1, \boldsymbol{X}_{i_1}, \ldots, \boldsymbol{X}_{i,m_i}\}$. We call a sequence of orthonormal ensembles $\boldsymbol{\mathcal{X}}$* independent *if the ensembles are independent families of random variables.*

In this section, we are only interested in independent sequences of orthonormal ensembles, and we will call these *sequences of ensembles* for brevity.

Let $q = 2^t$ for some $t \in \mathbb{N}$. Let $\boldsymbol{z}_i$ be a random variable uniformly distributed over $\{\pm 1\}^t$, and let $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N)$ be the random variable uniformly distributed over $\{\pm 1\}^{Nt}$.

Let $\boldsymbol{\mathcal{X}} = (\boldsymbol{\mathcal{X}}_1, \boldsymbol{\mathcal{X}}_2, \ldots, \boldsymbol{\mathcal{X}}_N)$ be a sequence of ensembles such that $\boldsymbol{\mathcal{X}}_i = \{\boldsymbol{X}_{i,0}, \boldsymbol{X}_{i_1}, \ldots, \boldsymbol{X}_{i,q-1}\}$ and $\boldsymbol{X}_{i,j} = \chi_j(\boldsymbol{z}_i)$, where $\{\chi_0 \equiv 1, \chi_1, \ldots, \chi_{q-1}\}$ is the set of characters for $\{\pm 1\}^t$.

Let $\boldsymbol{\mathcal{Y}} = (\boldsymbol{\mathcal{Y}}_1, \boldsymbol{\mathcal{Y}}_2, \ldots, \boldsymbol{\mathcal{Y}}_N)$ be a sequence of ensembles such that $\boldsymbol{\mathcal{Y}}_i = \{\boldsymbol{Y}_{i,0}, \boldsymbol{Y}_{i,1}, \ldots, \boldsymbol{Y}_{i,q-1}\}$ where $\boldsymbol{Y}_{i,0} \equiv 1$ and $\boldsymbol{Y}_{i,j}$ is an independent uniform sample from $\{\pm 1\}$ for $j > 0$.

Now we are ready to state our invariance principle for the fourth moment of f.

Theorem 9.1.2. *Let $\hat{f}(\alpha)$ be the indeterminates. Define $\mathbf{Inf}_i(f) = \sum_{\alpha:\alpha_i>0} \hat{f}(\alpha)^2$. For every constant $\beta > 0$, we have*

$$\left\{\sum_{\alpha} \hat{f}(\alpha)^2 \leq 1\right\} \vdash_4 \mathbf{E}\, f(\boldsymbol{\mathcal{X}})^4 \leq \mathbf{E}\, f(\boldsymbol{\mathcal{Y}})^4 + \beta \cdot q^{5\ell} + \left(1 + \frac{1}{\beta}\right) \sum_{i \in [N]} \mathbf{Inf}_i(f)^2.$$

Proof. Let $\boldsymbol{\mathcal{Z}}^{(i)} = (\boldsymbol{\mathcal{X}}_1, \ldots, \boldsymbol{\mathcal{X}}_i, \boldsymbol{\mathcal{Y}}_{i+1}, \ldots, \boldsymbol{\mathcal{Y}}_N)$ for $i = 0, 1, 2, \ldots, N$. Also for each $i = 1, 2, \ldots, N$, let

$$\mathrm{E}_i f(x) = \sum_{\alpha:\alpha_i=0} \hat{f}(\alpha) x_\alpha,$$

and

$$\mathrm{D}_i f(x) = \sum_{\alpha:\alpha_i>0} \hat{f}(\alpha) x_\alpha.$$

For all $i = 1, 2, \ldots, N$, we have the following polynomial identities,

$$\begin{aligned}
&\mathbf{E}\, f(\boldsymbol{\mathcal{Z}}^{(i)})^4 - \mathbf{E}\, f(\boldsymbol{\mathcal{Z}}^{(i-1)})^4 \\
=&\, \mathbf{E}\left[\mathrm{E}_i f(\boldsymbol{\mathcal{Z}}^{(i)}) + \mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i)})\right]^4 - \mathbf{E}\left[\mathrm{E}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)}) + \mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)})\right]^4
\end{aligned}$$

$$= \mathbf{E}\left[\mathrm{E}_i f(\boldsymbol{\mathcal{Z}}^{(i)})\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i)})^3 + \mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i)})^4\right] - \mathbf{E}\left[\mathrm{E}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)})\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)})^3 + \mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)})^4\right],$$

where in the second identity we use the fact that the first two moments of $\mathcal{X}_i$ and $\mathcal{Y}_i$ match.

Now, for every constant $\beta > 0$, we use Fact 8.3.10 and the simple fact that $\vdash_4 - \mathbf{E}\,\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)})^4 \leq 0$, and get

$$\begin{aligned}\vdash_4 \mathbf{E}\, f(\boldsymbol{\mathcal{Z}}^{(i)})^4 - \mathbf{E}\, f(\boldsymbol{\mathcal{Z}}^{(i-1)})^4 \\ \leq \mathbf{E}\left[\frac{\beta}{2}\mathrm{E}_i f(\boldsymbol{\mathcal{Z}}^{(i)})^2\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i)})^2 + \left(1 + \frac{1}{2\beta}\right)\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i)})^4\right] \\ + \mathbf{E}\left[\frac{\beta}{2}\mathrm{E}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)})^2\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)})^2 + \frac{1}{2\beta}\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)})^4\right].\end{aligned} \tag{9.1}$$

We view $\mathrm{E}_i f(\boldsymbol{\mathcal{Z}}^{(i)})$ as a multilinear polynomial over $\{\pm 1\}$ inputs with degree at most $t\ell$, and view $\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i)})$ as a multilinear polynomial over the same set of inputs with degree at most $t\ell$. Therefore, by hypercontractivity inequality (Theorem 8.4.3), we have

$$\begin{aligned}\vdash_4 \mathbf{E}\,\mathrm{E}_i f(\boldsymbol{\mathcal{Z}}^{(i)})^2\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i)})^2 &\leq 9^{t\ell}\left(\mathbf{E}\,\mathrm{E}_i f(\boldsymbol{\mathcal{Z}}^{(i)})^2\right)\left(\mathbf{E}\,\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i)})^2\right) \\ &= 9^{t\ell}\left(\mathbf{E}\,\mathrm{E}_i f(\boldsymbol{\mathcal{X}})^2\right)\left(\mathbf{E}\,\mathrm{D}_i f(\boldsymbol{\mathcal{X}})^2\right) = 9^{t\ell}\left(\mathbf{E}\,\mathrm{E}_i f(\boldsymbol{\mathcal{X}})^2\right)\mathbf{Inf}_i(f),\end{aligned} \tag{9.2}$$

where the first equality is because the first two moments of $\boldsymbol{\mathcal{Z}}^{(i)}$ match those of $\boldsymbol{\mathcal{X}}$, and the second one is by the definition of $\mathbf{Inf}_i(f)$. Similarly, we have

$$\begin{aligned}\vdash_4 \mathbf{E}\,\mathrm{E}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)})^2\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)})^2 &\leq 9^{t\ell}\left(\mathbf{E}\,\mathrm{E}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)})^2\right)\left(\mathbf{E}\,\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)})^2\right) \\ &= 9^{t\ell}\left(\mathbf{E}\,\mathrm{E}_i f(\boldsymbol{\mathcal{X}})^2\right)\left(\mathbf{E}\,\mathrm{D}_i f(\boldsymbol{\mathcal{X}})^2\right) = 9^{t\ell}\left(\mathbf{E}\,\mathrm{E}_i f(\boldsymbol{\mathcal{X}})^2\right)\mathbf{Inf}_i(f).\end{aligned} \tag{9.3}$$

We also view $\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i)})^4$ as a multilinear polynomial over $\{\pm 1\}$ inputs with degree at most $t\ell$. Using Theorem 8.4.3, we get

$$\vdash_4 \mathbf{E}\,\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i)})^4 \leq 9^{t\ell}\left(\mathbf{E}\,\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i)})^2\right)^2 = \left(\mathbf{E}\,\mathrm{D}_i f(\boldsymbol{\mathcal{X}})^2\right)^2 = \left(\mathbf{Inf}_i(f)\right)^2. \tag{9.4}$$

Again, the first equality is because the first two moments of $\boldsymbol{\mathcal{Z}}^{(i)}$ match those of $\boldsymbol{\mathcal{X}}$, while the second equality is by the definition of $\mathbf{Inf}_i(f)$. Similarly, we have

$$\vdash_4 \mathbf{E}\,\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)})^4 \leq 9^{t\ell}\left(\mathbf{E}\,\mathrm{D}_i f(\boldsymbol{\mathcal{Z}}^{(i-1)})^2\right)^2 = \left(\mathbf{E}\,\mathrm{D}_i f(\boldsymbol{\mathcal{X}})^2\right)^2 = \left(\mathbf{Inf}_i(f)\right)^2. \tag{9.5}$$

Now we incorporate (9.2), (9.3), (9.4), and (9.5) into (9.1), and get

$$\vdash_4 \mathbf{E}\, f(\boldsymbol{\mathcal{Z}}^{(i)})^4 - \mathbf{E}\, f(\boldsymbol{\mathcal{Z}}^{(i-1)})^4 \le \beta \cdot 9^{t\ell} \left(\mathbf{E}\, \mathrm{E}_i f(\boldsymbol{\mathcal{X}})^2\right) \mathbf{Inf}_i(f) + \left(1 + \frac{1}{\beta}\right) \left(\mathbf{Inf}_i(f)\right)^2.$$

Therefore, for every constant $\beta, > 0$,

$$\left\{\sum_\alpha \hat{f}(\alpha)^2 \le 1\right\} \vdash_4 \mathbf{E}\, f(\boldsymbol{\mathcal{Z}}^{(i)})^4 - \mathbf{E}\, f(\boldsymbol{\mathcal{Z}}^{(i-1)})^4 \le \beta \cdot 9^{t\ell} \cdot \mathbf{Inf}_i(f) + \left(1 + \frac{1}{\beta}\right) \left(\mathbf{Inf}_i(f)\right)^2. \tag{9.6}$$

Now we sum up (9.6) over all $i = 1, 2, \ldots, N$, and get

$$\left\{\sum_\alpha \hat{f}(\alpha)^2 \le 1\right\} \vdash_4 \mathbf{E}\, f(\boldsymbol{\mathcal{X}})^4 - \mathbf{E}\, f(\boldsymbol{\mathcal{Y}})^4 \le \beta \cdot 9^{t\ell} \cdot \sum_{i\in[N]} \mathbf{Inf}_i(f) + \left(1 + \frac{1}{\beta}\right) \sum_{i\in[N]} \left(\mathbf{Inf}_i(f)\right)^2$$
$$\le \beta\ell \cdot 9^{t\ell} + \left(1 + \frac{1}{\beta}\right) \sum_{i\in[N]} \left(\mathbf{Inf}_i(f)\right)^2 \le \beta \cdot q^{5\ell} + \left(1 + \frac{1}{\beta}\right) \sum_{i\in[N]} \left(\mathbf{Inf}_i(f)\right)^2.$$

□

The following corollary will be useful.

Corollary 9.1.3. *Let $\hat{f}(\alpha)$ be the indeterminates. Define $\mathbf{Inf}_i(f) = \sum_{\alpha:\alpha_i>0} \hat{f}(\alpha)^2$. For every constant $\beta > 0$, we have*

$$\left\{\sum_\alpha \hat{f}(\alpha)^2 \le 1\right\} \vdash_4 \mathbf{E}\, f(\boldsymbol{\mathcal{X}})^4 \le 9^\ell \left(\mathbf{E}\, f(\boldsymbol{\mathcal{X}})^2\right)^2 + \beta \cdot q^{5\ell} + \left(1 + \frac{1}{\beta}\right) \sum_{i\in[N]} \mathbf{Inf}_i(f)^2.$$

Proof. Since $f(\boldsymbol{\mathcal{Y}})$ is a multilinear polynomial over $\{\pm 1\}$ inputs with degree at most ℓ, by hypercontractivity inequality (Theorem 8.4.3), we have

$$\vdash_4 \mathbf{E}\, f(\boldsymbol{\mathcal{Y}})^4 \le 9^\ell \left(\mathbf{E}\, f(\boldsymbol{\mathcal{Y}})^2\right)^2.$$

We apply this together with the polynomial identity $\left(\mathbf{E}\, f(\boldsymbol{\mathcal{Y}})^2\right)^2 = \left(\mathbf{E}\, f(\boldsymbol{\mathcal{X}})^2\right)^2$ (since the first two moments of $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ match) to Theorem 9.1.2, and get the desired conclusion. □

9.2 Analysis of the dictatorship test gadget for UNIQUEGAMES in the SOS proof system

In this section, we are going to prove the following theorem (Theorem 9.2.1) in the SOS proof system. The theorem can be viewed as an analog of the "Majority Is Stablest" theorem in [174], and serves the same purpose – to show that any function with no influential coordinates succeeds with very small probability in the dictatorship test for UNIQUEGAMES.

Let $q = t^t$ for some $t \in \mathbb{N}$ throughout this section.

Theorem 9.2.1. *Let the entries in $f(x)$ where $x \in \mathbb{Z}_q^N$ be indeterminates. Let T_ρ be the operator on $f : \mathbb{Z}_q^N \to \mathbb{R}$ such that $\mathrm{T}_\rho f(x) = \mathbf{E}_{\boldsymbol{y} \sim_\rho x} f(\boldsymbol{y})$, where $\boldsymbol{y} \sim_\rho x$ means each $\boldsymbol{y}_i$ independently takes the value x_i with probability ρ, and takes a uniform random value in $\mathbb{Z}_q$ with probability $(1-\rho)$. Define $\langle f, \mathrm{T}_\rho f \rangle = \mathbf{E}_{\boldsymbol{x} \in F_q^N} f(\boldsymbol{x}) \mathrm{T}_\rho f(\boldsymbol{x})$.*

For all constants $\delta, \gamma \in [0,1]$ and $\beta > 0$, we have

$$\left\{ f(x)^2 \le f(x) : \forall x, \mathbf{E} f \le \delta \right\} \vdash_4$$
$$\langle f, \mathrm{T}_{1-\gamma} f \rangle \le \delta^{1+.01\gamma} + \delta^{1.1} + \frac{1}{4\delta^{.3}} \left(\beta \cdot q^{.1 \log \frac{1}{\delta}} + \left(1 + \frac{1}{\beta}\right) \sum_{i \in [N]} \mathbf{Inf}_i^{\le .02 \log \frac{1}{\delta}} (f)^2 \right).$$

Proof. For every constant $\lambda : 0 < \lambda < 1$, let $\mathrm{P}_{>\lambda}$ be the operator so that $\mathrm{P}_{>\lambda} f$ is the projection of f on to the eigenspace of $\mathrm{T}_{1-\gamma}$ with eigenvalue greater than λ. The following SOS statement is easy to deduce.

$$\vdash_2 \langle f, \mathrm{T}_{1-\gamma} f \rangle \le \langle f, \mathrm{P}_{>\lambda} f \rangle + \lambda \, \mathbf{E} f^2. \tag{9.7}$$

On the other hand, by Fact 8.3.13, we know that for every constant $\epsilon > 0$, we have

$$\left\{ f(x)^2 \le f(x) : \forall x \right\} \vdash_4 \langle f, \mathrm{P}_{>\lambda} f \rangle \le \frac{3\epsilon}{4} \mathbf{E} f + \frac{1}{4\epsilon^3} \mathbf{E} \left(\mathrm{P}_{>\lambda} f \right)^4. \tag{9.8}$$

Observe that both f and $\mathrm{P}_{>\lambda} f$ can be written in the multilinear forms of

$$f(x) = \sum_{\alpha \in \mathbb{N}^N} \hat{f}(\alpha) \prod_{i=1}^n \chi_{\alpha_i}(x_i)$$
$$\mathrm{P}_{>\lambda} f(x) = \sum_{\alpha \in \mathbb{N}^N, |\alpha| \le \frac{\ln \lambda}{\ln(1-\gamma)}} \hat{f}(\alpha) \prod_{i=1}^n \chi_{\alpha_i}(x_i),$$

where $\{\chi_0 \equiv 1, \chi_1, \ldots, \chi_{q-1}\}$ is the set of characters for $\{\pm 1\}^t$. Also, $\hat{f}(\alpha)$ can be written as linear combinations of the indeterminates $f(x)$; and $\{f(x)^2 \leq f(x) : \forall x, \mathbf{E} f \leq \delta\} \vdash_2 \mathbf{E} f^2 \leq 1$ (when $\delta \leq 1$), and $\{\mathbf{E} f^2 \leq 1\} \vdash_2 \sum_\alpha \widehat{\mathrm{P}_{>\lambda} f}(\alpha)^2 \leq 1$. Therefore, let

$$\ell \stackrel{\text{def}}{=} \frac{2}{\gamma} \log \frac{1}{\lambda}$$

which upper bounds the degree of $\mathrm{P}_{>\lambda} f$, by Corollary 9.1.3, for every constant $\beta > 0$,

$$\left\{f(x)^2 \leq f(x) : \forall x, \mathbf{E} f \leq \delta\right\} \vdash_4 9^\ell \left(\mathbf{E}\left(\mathrm{P}_{>\lambda} f\right)^2\right)^2 + \beta \cdot q^{5\ell} + \left(1 + \frac{1}{\beta}\right) \sum_{i \in [N]} \mathbf{Inf}_i^{\leq \ell}(f)^2, \tag{9.9}$$

where $\mathbf{Inf}_i^{\leq \ell}(f) = \sum_{\alpha, |\alpha| \leq \ell} \hat{f}(\alpha)^2$ is the low-degree influence of f. Since

$$\vdash_4 \left(\mathbf{E}\left(\mathrm{P}_{>\lambda} f\right)^2\right)^2 \leq \left(\mathbf{E} f^2\right)^2,$$

and

$$\left\{f(x)^2 \leq f(x) : \forall x\right\} \vdash_2 \mathbf{E} f^2 \leq \mathbf{E} f, \tag{9.10}$$

together with (9.8) and (9.9), we get for all constants $\epsilon, \beta > 0$,

$$\begin{aligned}
&\left\{f(x)^2 \leq f(x) : \forall x, \mathbf{E} f \leq \delta\right\} \vdash_4 \\
&\qquad \langle f, \mathrm{P}_{>\lambda} f\rangle \leq \frac{3\epsilon}{4} \mathbf{E} f + \frac{9^\ell}{4\epsilon^3} (\mathbf{E} f)^2 + \frac{\beta \cdot q^{5\ell}}{4\epsilon^3} + \frac{1}{4\epsilon^3}\left(1 + \frac{1}{\beta}\right) \sum_{i \in [N]} \mathbf{Inf}_i^{\leq \ell}(f)^2 \\
&\qquad\qquad \leq \frac{3\epsilon\delta}{4} + \frac{9^\ell \delta^2}{4\epsilon^3} + \frac{\beta \cdot q^{5\ell}}{4\epsilon^3} + \frac{1}{4\epsilon^3}\left(1 + \frac{1}{\beta}\right) \sum_{i \in [N]} \mathbf{Inf}_i^{\leq \ell}(f)^2.
\end{aligned} \tag{9.11}$$

We combine (9.11), (9.10), and (9.7), and get

$$\begin{aligned}
&\left\{f(x)^2 \leq f(x) : \forall x, \mathbf{E} f \leq \delta\right\} \vdash_4 \\
&\qquad \langle f, \mathrm{T}_{1-\gamma} f\rangle \leq \lambda\delta + \frac{3\epsilon\delta}{4} + \frac{9^\ell \delta^2}{4\epsilon^3} + \frac{\beta \cdot q^{5\ell}}{4\epsilon^3} + \frac{1}{4\epsilon^3}\left(1 + \frac{1}{\beta}\right) \sum_{i \in [N]} \mathbf{Inf}_i^{\leq \ell}(f)^2.
\end{aligned} \tag{9.12}$$

Now we set $\lambda = \delta^{.01\gamma}$ (and therefore $\ell = .02 \log \frac{1}{\delta}$), and $\epsilon = \delta^{.1}$. We have $\frac{3\epsilon\delta}{4} + \frac{9^\ell \delta^2}{4\epsilon^3} \leq \delta^{1.1}$. Therefore, (9.12) implies

$$\left\{f(x)^2 \le f(x) : \forall x, \mathbf{E}\, f \le \delta\right\} \vdash_4$$

$$\langle f, \mathrm{T}_{1-\gamma} f\rangle \le \delta^{1+.01\gamma} + \delta^{1.1} + \frac{1}{4\delta^{.3}}\left(\beta \cdot q^{.1\log\frac{1}{\delta}} + \left(1+\frac{1}{\beta}\right)\sum_{i\in[N]} \mathbf{Inf}_i^{\le .02\log\frac{1}{\delta}}(f)^2\right)$$

for every constant $\beta > 0$. □

9.3 The KV UNIQUEGAMES instance and its SOS analysis

Let us recall the UNIQUEGAMES problem with label-size $N \in \mathbb{N}^+$. Given is a regular weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (self-loops allowed) with weights summing to 1. Also, given for each edge (u, v) is a permutation $\pi_{uv} : [N] \to [N]$. We write $(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\pi}) \sim \mathcal{E}$ to denote that edge $(\boldsymbol{u}, \boldsymbol{v})$ with pumutation $\boldsymbol{\pi} = \pi_{\boldsymbol{uv}}$ is chosen with probability equal to its edge weight. The goal is to give a labeling $F : \mathcal{V} \to [N]$ so as to maximize $\Pr_{(\boldsymbol{u},\boldsymbol{v},\boldsymbol{\pi})\sim\mathcal{E}}[\boldsymbol{\pi}(F(\boldsymbol{u})) = F(\boldsymbol{v})]$. The natural polynomial optimization formulation has an indeterminate $X_{u,i}$ for each $u \in \mathcal{V}, i \in [k]$:

$$\begin{aligned}
\max \quad & \mathop{\mathbf{E}}_{(\boldsymbol{u},\boldsymbol{v},\boldsymbol{\pi})\sim\mathcal{E}}\Big[\sum_{i=1}^{N} X_{\boldsymbol{u},i} X_{\boldsymbol{v},\boldsymbol{\pi}(i)}\Big] = \mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}}\Big[\sum_{i=1}^{N} X_{\boldsymbol{u},i} \cdot \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim\boldsymbol{u}}[X_{\boldsymbol{v},\boldsymbol{\pi}(i)}]\Big] \\
\text{s.t.} \quad & X_{u,i}^2 = X_{u,i} \quad \forall u \in \mathcal{V}, i \in [N] \\
& \sum_{i=1}^{N} X_{u,i} = 1 \quad \forall u \in \mathcal{V},
\end{aligned}$$

where we write $(\boldsymbol{v}, \boldsymbol{\pi}) \sim u$ in place of $(u, \boldsymbol{v}, \boldsymbol{\pi}) \sim \mathcal{E}|_{\boldsymbol{u}=u}$ for brevity. Thus the degree-d SOS SDP hierarchy will use binary search to compute the smallest β for which

$$\begin{aligned}
\{X_{u,i}^2 = X_{u,i} : \forall u \in \mathcal{V}, i \in [N]\} \cup \{\sum_{i=1}^{N} X_{u,i} = 1 : \forall u \in \mathcal{V}\} \\
\cup \left\{\mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}}\Big[\sum_{i=1}^{N} X_{\boldsymbol{u},i} \cdot \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim\boldsymbol{u}}[X_{\boldsymbol{v},\boldsymbol{\pi}(i)}]\Big] \ge \beta\right\} \quad \vdash_d \quad -1 \ge 0. \quad (9.13)
\end{aligned}$$

The Khot–Vishnoi UNIQUEGAMES instance is defined as follows. Fix $k \in \mathbb{N}$. Let $\mathcal{F} = \{\{\pm 1\}^k \to \{\pm 1\}\}$ be the family of Boolean functions on $\{\pm 1\}^k$. Consider the following equivalence relation $\equiv$ on F. For any two Boolean functions $f, g \in \mathcal{F}$, we say $f \equiv g$ if and only if there is an $S \subseteq [k]$ such that $f = g\chi_S$ where $\chi_S(i) = \prod_{i\in S} x_i$. Now this relation partitions $\mathcal{F}$ into equivalence classes $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_m$. We denote by $[\mathcal{P}_i]$ an arbitrary representative in $\mathcal{P}_i$. For each $f \in \mathcal{F}$, let $\mathcal{P}(f)$ be the $\mathcal{P}_i$ which contains f. Now we are ready to describe the UNIQUEGAMES instance $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with parameter $\eta > 0$. The vertex set $\mathcal{V}$ is simply the set of equivalence classes. The label set is $[N]$. For each vertex $\mathcal{P}_i$, we decide an arbitrary one-to-one correspondence between the elements in $\mathcal{P}_i$ and $[N]$ so that the permutations $\boldsymbol{\pi}$ can be described as mappings between the elements in $\mathcal{P}_i$ and $\mathcal{P}_j$. We define the edge set together with the weight distribution $(\mathcal{P}_{\boldsymbol{i}}, \mathcal{P}_{\boldsymbol{j}}, \boldsymbol{\pi}) \sim \mathcal{E}$ as follows,

- choose $\boldsymbol{f}$ as an uniform random function in $\mathcal{F}$, and let $\mathcal{P}_{\boldsymbol{i}} = \mathcal{P}(\boldsymbol{f})$;
- choose $\boldsymbol{g} \sim_{1-\eta} \boldsymbol{f}$, i.e. we obtain $\boldsymbol{g}$ by flipping each entry of $\boldsymbol{f}$ independently with probability 2η. Let $\mathcal{P}_{\boldsymbol{j}} = \mathcal{P}(\boldsymbol{g})$;
- finally, since $\mathcal{P}_{\boldsymbol{i}} = \{\boldsymbol{f}\chi_S : \forall S \subseteq [k]\}$ and $\mathcal{P}_{\boldsymbol{j}} = \{\boldsymbol{g}\chi_S : \forall S \subseteq [k]\}$, we define $\boldsymbol{\pi} : \mathcal{P}_{\boldsymbol{i}} \to \mathcal{P}_{\boldsymbol{j}}$ as $\boldsymbol{\pi}(\boldsymbol{f}\chi_S) = \boldsymbol{g}\chi_S$ for every $S \subseteq [k]$.

We are going to show degree-4 SOS proofs to the statement that the UNIQUEGAMES instance $\mathcal{G}$ defined as above does not have great solution. In fact we are going to show something stronger than (9.13), in the sense that we only assume $X_{u,i}^2 \le X_{u_i}$ and $\mathbf{E}_{u\in\mathcal{V}} \mathbf{E}_{i\in[N]} X_{u,i} \le 1/N$, and we will give an SOS proof instead of an SOS refutation. We prove

Theorem 9.3.1. *Given $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as the instance described above, for every constant $\delta \in (0,1)$, we have*

$$\{X_{u,i}^2 \le X_{u,i} : \forall u \in \mathcal{V}, i \in [N]\} \cup \{\mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}} \mathop{\mathbf{E}}_{\boldsymbol{i}\in[N]} X_{\boldsymbol{u},\boldsymbol{i}} \le \delta\} \vdash_4$$
$$\mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}} \Big[\sum_{i=1}^{N} X_{\boldsymbol{u},i} \cdot \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim\boldsymbol{u}} [X_{\boldsymbol{v},\boldsymbol{\pi}(i)}]\Big] \le N \cdot \delta^{1+\Omega(\eta)}.$$

Therefore, when $\delta = 1/N$, we have

$$\{X_{u,i}^2 \le X_{u,i} : \forall u \in \mathcal{V}, i \in [N]\} \cup \{\mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}} \mathop{\mathbf{E}}_{\boldsymbol{i}\in[N]} X_{\boldsymbol{u},\boldsymbol{i}} \le \tfrac{1}{N}\} \vdash_4$$
$$\mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}} \Big[\sum_{i=1}^{N} X_{\boldsymbol{u},i} \cdot \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim\boldsymbol{u}} [X_{\boldsymbol{v},\boldsymbol{\pi}(i)}]\Big] \le N^{-\Omega(\eta)}.$$

Proof. We first perform some notational manipulations in order to make the representation easier. Observe that each pair $(u,i) \in \mathcal{V} \times [N]$ is uniquely mapped to a function $f \in \mathcal{F}$. Therefore, we will talk about $X(f) : f \in \mathcal{F}$ instead of $X_{u,i} : (u,i) \in \mathcal{V} \times [N]$. Given

$u \in \mathcal{V}$ and $i \in [N]$ (therefore $f \in \mathcal{F}$), let $\boldsymbol{g} \in \mathcal{F}$ be the function corresponding to $(\boldsymbol{v}, \boldsymbol{\pi}(i))$ where $(\boldsymbol{v}, \boldsymbol{\pi}) \sim \boldsymbol{u}$. Observe that $\boldsymbol{g}$ in fact follows the distribution $\boldsymbol{g} \sim_{1-\eta} f$, and therefore

$$\mathop{\mathbf{E}}_{\boldsymbol{u} \in \mathcal{V}} \left[\sum_{i=1}^{N} X_{\boldsymbol{u},i} \cdot \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi}) \sim \boldsymbol{u}} [X_{\boldsymbol{v},\boldsymbol{\pi}(i)}] \right] = N \mathop{\mathbf{E}}_{\boldsymbol{f} \in \mathcal{F}, \boldsymbol{g} \sim_{1-\eta} \boldsymbol{f}} X(\boldsymbol{f}) X(\boldsymbol{g}) \tag{9.14}$$

if we assign $X(f)$ according to $X_{u,i}$.

Therefore, to prove the theorem, it suffices to show that

$$\{X(f)^2 \le X(f) : \forall f \in \mathcal{F}\} \cup \{\mathop{\mathbf{E}}_{\boldsymbol{f} \in \mathcal{F}} X(\boldsymbol{f}) \le \delta\} \vdash_4 \langle X, \mathrm{T}_{1-\eta} X \rangle \le \delta^{1+\Omega(\eta)}. \tag{9.15}$$

where $\mathrm{T}_{1-\eta}$ is the operator so that $\mathrm{T}_{1-\eta} X(f) = \mathbf{E}_{\boldsymbol{g} \sim_{1-\eta} f} X(\boldsymbol{g})$ and we define $\langle X, \mathrm{T}_{1-\eta} X \rangle = \mathbf{E}_{\boldsymbol{f} \in \mathcal{F}} X(\boldsymbol{f}) \mathrm{T}_{1-\eta} X(\boldsymbol{f})$.

The rest of the proof is to prove (9.15). For every $d \in \mathbb{N}$, it is easy to show that

$$\vdash_2 \langle X, \mathrm{T}_{1-\eta} X \rangle \le \langle X, \mathrm{P}_{\ge (1-\eta)^d} X \rangle + (1-\eta)^d \mathop{\mathbf{E}}_{\boldsymbol{f} \in \mathcal{F}} X(\boldsymbol{f})^2, \tag{9.16}$$

where $\mathrm{P}_{\ge (1-\eta)^d}$ is the operator so that $\mathrm{P}_{\ge (1-\eta)^d} X$ is the projection of X on to the span of the characters χ_S where $|S| \le d$. Now we apply Fact 8.3.13 to (9.16), and have that for every constant $\epsilon > 0$, we have

$$\vdash_4 \langle X, \mathrm{T}_{1-\eta} X \rangle \le \frac{3\epsilon}{4} \mathop{\mathbf{E}}_{\boldsymbol{f} \in \mathcal{F}} X(\boldsymbol{f}) + \frac{1}{4\epsilon^3} \mathop{\mathbf{E}}_{\boldsymbol{f} \in \mathcal{F}} (\mathrm{P}_{\ge (1-\eta)^d} X(\boldsymbol{f}))^4 + (1-\eta)^d \mathop{\mathbf{E}}_{\boldsymbol{f} \in \mathcal{F}} X(\boldsymbol{f})^2. \tag{9.17}$$

Observe that $\mathrm{P}_{\ge (1-\eta)^d} X(f)$ is a multilinear polynomial of $f(x)$ with degree at most d. Therefore, by hypercontractivity inequality (Theorem 8.4.3) we have

$$\vdash_4 \langle X, \mathrm{T}_{1-\eta} X \rangle \le \frac{3\epsilon}{4} \mathop{\mathbf{E}}_{\boldsymbol{f} \in \mathcal{F}} X(\boldsymbol{f}) + \frac{9^d}{4\epsilon^3} \left(\mathop{\mathbf{E}}_{\boldsymbol{f} \in \mathcal{F}} (\mathrm{P}_{\ge (1-\eta)^d} X(\boldsymbol{f}))^2 \right)^2 + (1-\eta)^d \mathop{\mathbf{E}}_{\boldsymbol{f} \in \mathcal{F}} X(\boldsymbol{f})^2. \tag{9.18}$$

With some simple SOS facts, (9.18) implies

$$\{X(f)^2 \le X(f) : \forall f \in \mathcal{F}\} \cup \{\mathop{\mathbf{E}}_{\boldsymbol{f} \in \mathcal{F}} X(\boldsymbol{f}) \le \delta\} \vdash_4$$
$$\langle X, \mathrm{T}_{1-\eta} X \rangle \le \delta \left(\frac{3\epsilon}{4} + (1-\eta)^d \right) + \frac{9^d \delta^2}{4\epsilon^3}$$

for every $\epsilon > 0$ and $d \in \mathbb{N}$.

We choose $d = \left\lceil \frac{\ln(1/\delta)}{\ln 81} \right\rceil$ and $\epsilon = N^{-.1}$ to establish (9.15). $\square$

Corollary 9.3.2. *Given $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as the instance in Theorem 9.3.1, for every constant $\delta \in (0, 1)$, we have*

$$\{X_{u,i}^2 \le X_{u,i} : \forall u \in \mathcal{V}, i \in [N]\} \cup \{\mathop{\mathbf{E}}_{u \in \mathcal{V}} \mathop{\mathbf{E}}_{i \in [N]} X_{u,i} \le \delta\} \vdash_4$$
$$\mathop{\mathbf{E}}_{\boldsymbol{u} \in \mathcal{V}} \left[\sum_{i=1}^{N} \left(\mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi}) \sim \boldsymbol{u}} [X_{\boldsymbol{v},\boldsymbol{\pi}(i)}] \right)^2 \right] \le N \cdot \delta^{1+\Omega(\eta)}.$$

Proof. The proof of this corollary is almost the same as that of Theorem 9.3.1. The only difference is that in (9.14), instead we observe that

$$\mathop{\mathbf{E}}_{\boldsymbol{u} \in \mathcal{V}} \left[\sum_{i=1}^{N} \left(\mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi}) \sim \boldsymbol{u}} [X_{\boldsymbol{v},\boldsymbol{\pi}(i)}] \right)^2 \right] = N \mathop{\mathbf{E}}_{\boldsymbol{f} \in \mathcal{F}} \left(\mathop{\mathbf{E}}_{\boldsymbol{g} \sim_{1-\eta} \boldsymbol{f}} X(\boldsymbol{g}) \right)^2,$$

and in (9.15), we turn to prove

$$\{X(f)^2 \le X(f) : \forall f \in \mathcal{F}\} \cup \{\mathop{\mathbf{E}}_{\boldsymbol{f} \in \mathcal{F}} X(\boldsymbol{f}) \le \tfrac{1}{N}\} \vdash_4$$
$$\langle \mathrm{T}_{1-\eta} X, \mathrm{T}_{1-\eta} X \rangle = \langle X, \mathrm{T}_{(1-\eta)^2} X \rangle \le \delta^{1+\Omega(\eta)}.$$

instead.

In the rest of the proof, we use $(1-\eta)^2$ whenever $(1-\eta)$ occurs, and reach the same conclusion.

□

9.4 Influence decoding and putting everything together

Now let us recall that the UNIQUEGAMES instances we are interested in Theorem 9.0.8 are obtained by composing the KKMO "noise stability" reduction from [141] with the KV integrality gap instances analyzed in the previous section. Let us fix an UNIQUEGAMES instance $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with label-size N (in this section, we are interested in the KV integrality gap instance presented in the previous section with parameter η). The KKMO reduction, parameterized by $\gamma \in [0, 1]$, creates a new UNIQUEGAMES instance $G = (V, E)$ with label-size q. In the vertex set V, there is a vertex $w_{u,x}$ for each $u \in \mathcal{V}$ and each $x \in \mathbb{Z}_q^N$. For each $c \in \mathbb{Z}_q$, let $\sigma_c(x) = x + c$. The probability distribution E on edges and permutations is given as follows.

- draw $\boldsymbol{u} \in \mathcal{V}$ uniformly;
- independently draw $(\boldsymbol{u}, \boldsymbol{v}_1, \boldsymbol{\pi}_1)$ and $(\boldsymbol{u}, \boldsymbol{v}_2, \boldsymbol{\pi}_2)$ from the marginal of $\mathcal{E}$ which is first vertex $\boldsymbol{u}$;
- draw $\boldsymbol{x} \in \mathbb{Z}_q^N$ uniformly;
- draw $\boldsymbol{y} \sim_{1-\gamma} \boldsymbol{x}$, i.e. each $\boldsymbol{y}_i$ independently becomes $\boldsymbol{x}_i$ with probability $1-\gamma$ and a uniform random element from $\mathbb{Z}_q$ with probability γ;
- pick a random element $\boldsymbol{c} \in \mathbb{Z}_q$;
- output the edge $(w_{\boldsymbol{u}_1, \boldsymbol{x} \circ \boldsymbol{\pi}_1}, w_{\boldsymbol{u}_2, (\boldsymbol{y}-\boldsymbol{c}) \circ \boldsymbol{\pi}_2})$ together with $\sigma_{\boldsymbol{c}}$ as the permutation.

Here $x \circ \pi$ is defined to be the string in $\mathbb{Z}_q^N$ such that $(x \circ \pi)_i = x_{\pi(i)}$ for all $i \in [N]$; and $y - c$ is the string in $\mathbb{Z}_q^N$ such that $(y-c)_i = y_i - c$ for all $i \in [N]$.

In this section we will prove the following theorem using the tools developed in the previous sections. The theorem directly implies Theorem 9.0.8 (when q is sufficiently large and $\log N \gg (\log q)^2/\eta$).

Theorem 9.4.1. *For each $u \in \mathcal{V}, x \in \mathbb{Z}_q^N, a \in \mathbb{Z}_q$, let $f_u^{(a)}(x)$ be the indeterminate which is intended to be the indicate variable for the event that $w_{u,x}$ takes the label a. Let*

$$A = \left\{ f_u^{(a)}(x)^2 = f_u^{(a)}(x) : \forall u \in \mathcal{V}, x \in \mathbb{Z}_q^N, a \in \mathbb{Z}_q \right\} \cup \left\{ \sum_{a \in \mathbb{Z}_q} f_u^{(a)}(x) = 1 : \forall u \in \mathcal{V}, x \in \mathbb{Z}_q^N \right\}.$$

We have

$$A \vdash_8 \mathsf{val}(f) \stackrel{\text{def}}{=} \mathop{\mathbf{E}}_{\boldsymbol{u} \in \mathcal{V}} \mathop{\mathbf{E}}_{\substack{(\boldsymbol{v}_1, \boldsymbol{\pi}_1) \sim \boldsymbol{u} \\ (\boldsymbol{v}_2, \boldsymbol{\pi}_2) \sim \boldsymbol{u}}} \mathop{\mathbf{E}}_{\substack{\boldsymbol{x} \in \mathbb{Z}_q^N \\ \boldsymbol{y} \sim_{1-\gamma} \boldsymbol{x}}} \mathop{\mathbf{E}}_{\boldsymbol{c} \in \mathbb{Z}_q} \sum_{a \in \mathbb{Z}_q} f_{\boldsymbol{v}_1}^{(a)}(\boldsymbol{x} \circ \boldsymbol{\pi}_1) f_{\boldsymbol{v}_2}^{(a+\boldsymbol{c})}((\boldsymbol{y} - \boldsymbol{c}) \circ \boldsymbol{\pi}_2)$$
$$\leq q^{-\Omega(\gamma)} + q^{O(\log q)} N^{-\Omega(\eta)}.$$

Proof. For each $u \in \mathcal{V}, x \in \mathbb{Z}_q^N, a \in \mathbb{Z}_q$, let $\tilde{f}_u^{(a)}(x) = \mathbf{E}_{\boldsymbol{c} \in \mathbb{Z}_q} f_u^{(a+\boldsymbol{c})}(x - c)$; let $h_u^{(a)}(x) = \mathbf{E}_{(\boldsymbol{v}, \boldsymbol{\pi}) \sim u} \tilde{f}_{\boldsymbol{v}}^{(a)}(x \circ \boldsymbol{\pi}^{-1})$. Let $\mathrm{T}_{1-\gamma}$ be the operator such that $\mathrm{T}_{1-\gamma} h(x) = \mathbf{E}_{\boldsymbol{y} \sim_{1-\gamma} x} h(\boldsymbol{y})$ when h is a function defined over $\mathbb{Z}_q^N$. The following polynomial identity is easy to verify.

$$\mathsf{val}(f) = \sum_{a \in \mathbb{Z}_q} \mathop{\mathbf{E}}_{\boldsymbol{u} \in \mathcal{V}} \langle h_{\boldsymbol{u}}^{(a)}, \mathrm{T}_{1-\gamma} h_{\boldsymbol{u}}^{(a)} \rangle.$$

Therefore, to prove the theorem, we only need to show that for every $a \in \mathbb{Z}_q$,

$$A \vdash_4 \mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}} \langle h_{\boldsymbol{u}}^{(a)}, \mathrm{T}_{1-\gamma} h_{\boldsymbol{u}}^{(a)} \rangle \le q^{-1-\Omega(\gamma)} + q^{O(\log q)} N^{-\Omega(\eta)}. \tag{9.19}$$

Now fix an arbitrary $a \in \mathbb{Z}_q$, by the definition of h (and using Fact 8.3.15) we have

$$\forall u \in \mathcal{V}, x \in \mathbb{Z}_q^n, A \vdash_2 h_u^{(a)}(x)^2 \le h_u^{(a)}(x), \tag{9.20}$$

and

$$\forall u \in \mathcal{V}, A \vdash_2 \mathop{\mathbf{E}}_{\boldsymbol{x}\in\mathbb{Z}_q^n} h_u^{(a)}(\boldsymbol{x}) = \frac{1}{q}. \tag{9.21}$$

Therefore, by Theorem 9.2.1, for every $u \in \mathcal{V}$ and every constant $\beta > 0$, we have [2]

$$A \vdash_4 \langle h_u^{(a)}, \mathrm{T}_{1-\gamma} h_u^{(a)} \rangle \le \frac{1}{q^{1+\Omega(\gamma)}} + \frac{1}{4q^{.3}} \left(\beta \cdot q^{.1 \log q} + \left(1 + \frac{1}{\beta}\right) \sum_{i\in[N]} \mathbf{Inf}_i^{\le .02 \log q}(h_u^{(a)})^2 \right). \tag{9.22}$$

Therefore, for every constant $\beta > 0$,

$$\begin{aligned} A \vdash_4 & \mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}} \langle h_{\boldsymbol{u}}^{(a)}, \mathrm{T}_{1-\gamma} h_{\boldsymbol{u}}^{(a)} \rangle \\ & \le \frac{1}{q^{1+\Omega(\gamma)}} + \frac{1}{4q^{.3}} \left(\beta \cdot q^{.1 \log q} + \left(1 + \frac{1}{\beta}\right) \mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}} \sum_{i\in[N]} \mathbf{Inf}_i^{\le .02 \log q}(h_{\boldsymbol{u}}^{(a)})^2 \right). \end{aligned} \tag{9.23}$$

Since (for every $\ell \in \mathbb{N}$)

$$\mathbf{Inf}_i^{\le \ell}(h_u^{(a)}) = \sum_{\alpha:|\alpha|\le \ell, \alpha_i > 0} \widehat{h_u^{(a)}}(\alpha)^2 = \sum_{\alpha:|\alpha|\le \ell, \alpha_i > 0} \left(\mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim u} \widehat{\tilde{f}_{\boldsymbol{v}}^{(a)}}(\alpha \circ \boldsymbol{\pi}) \right)^2,$$

by Fact 8.3.15, we have

$$\vdash_2 \mathbf{Inf}_i^{\le \ell}(h_u^{(a)}) \le \sum_{\alpha:|\alpha|\le \ell, \alpha_i > 0} \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim u} \widehat{\tilde{f}_{\boldsymbol{v}}^{(a)}}(\alpha \circ \boldsymbol{\pi})^2 = \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim u} \mathbf{Inf}_{\boldsymbol{\pi}(i)}^{\le \ell}(\tilde{f}_{\boldsymbol{v}}^{(a)}). \tag{9.24}$$

[2]This would require degree 6 if we apply Fact 8.2.3 directly. However, if carefully examining the details, one can see that degree-4 SOS proof suffices.

Therefore,

$$\vdash_4 \mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}} \sum_{i\in[N]} \mathbf{Inf}_i^{\le .02\log q}(h_{\boldsymbol{u}}^{(a)})^2 \le \sum_{i\in[N]} \mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}} \left(\mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim\boldsymbol{u}} \mathbf{Inf}_{\boldsymbol{\pi}(i)}^{\le .02\log q}(\tilde{f}_{\boldsymbol{v}}^{(a)}) \right)^2 \tag{9.25}$$

On the other hand, for every $i \in [N], v \in \mathcal{V}$, we have the following proof (whose proof will be deferred to the end of this section).

Claim 9.4.2. $A \vdash_4 \mathbf{Inf}_i^{\le .02\log q}(\tilde{f}_v^{(a)})^2 \le \mathbf{Inf}_i^{\le .02\log q}(\tilde{f}_v^{(a)})$.

Since

$$\mathop{\mathbf{E}}_{\boldsymbol{v}\in\mathcal{V}} \mathop{\mathbf{E}}_{\boldsymbol{i}\in[N]} \mathbf{Inf}_i^{\le .02\log q} \tilde{f}_{\boldsymbol{v}}^{(a)} = \frac{1}{N} \mathop{\mathbf{E}}_{\boldsymbol{v}\in\mathcal{V}} \sum_{\alpha:|\alpha|\le .02\log q} |\alpha| \widehat{\tilde{f}_{\boldsymbol{v}}^{(a)}}(\alpha)^2,$$

We also have

$$\vdash_2 \mathop{\mathbf{E}}_{\boldsymbol{v}\in\mathcal{V}} \mathop{\mathbf{E}}_{\boldsymbol{i}\in[N]} \mathbf{Inf}_i^{\le .02\log q} \tilde{f}_{\boldsymbol{v}}^{(a)} \le \frac{.02\log q}{N} \mathop{\mathbf{E}}_{\boldsymbol{v}\in\mathcal{V}} \mathop{\mathbf{E}}_{\boldsymbol{x}} \tilde{f}_{\boldsymbol{v}}^{(a)}(\boldsymbol{x})^2. \tag{9.26}$$

It's easy to deduce that $\vdash_2 \tilde{f}_v^{(a)}(x)^2 \le \tilde{f}_v^{(a)}(x)$ (for all $v \in \mathcal{V}, x \in \mathbb{Z}_q^N$). Therefore, (9.26) implies

$$\vdash_2 \mathop{\mathbf{E}}_{\boldsymbol{v}\in\mathcal{V}} \mathop{\mathbf{E}}_{\boldsymbol{i}\in[N]} \mathbf{Inf}_i^{\le .02\log q} \tilde{f}_{\boldsymbol{v}}^{(a)} \le \frac{.02\log q}{N} \mathop{\mathbf{E}}_{\boldsymbol{v}\in\mathcal{V}} \mathop{\mathbf{E}}_{\boldsymbol{x}} \tilde{f}_{\boldsymbol{v}}^{(a)}(\boldsymbol{x}) = \frac{.02\log q}{N} \mathop{\mathbf{E}}_{\boldsymbol{v}\in\mathcal{V}} \mathop{\mathbf{E}}_{\boldsymbol{x}} \mathop{\mathbf{E}}_{\boldsymbol{a}} f_{\boldsymbol{v}}^{(\boldsymbol{a})}(\boldsymbol{x}). \tag{9.27}$$

Therefore,

$$A \vdash_2 \mathop{\mathbf{E}}_{\boldsymbol{v}\in\mathcal{V}} \mathop{\mathbf{E}}_{\boldsymbol{i}\in[N]} \mathbf{Inf}_i^{\le .02\log q} \tilde{f}_{\boldsymbol{v}}^{(a)} \le \frac{.02\log q}{Nq}. \tag{9.28}$$

Claim 9.4.2, (9.28), Corollary 9.3.2, and Fact 8.2.3 imply

$$A \vdash_8 \sum_{i\in[N]} \mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}} \left(\mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim\boldsymbol{u}} \mathbf{Inf}_{\boldsymbol{\pi}(i)}^{\le .02\log q}(\tilde{f}_{\boldsymbol{v}}^{(a)}) \right)^2 \le N \left(\frac{.02\log q}{Nq} \right)^{1+\Omega(\eta)} \le \frac{\log q}{N^{\Omega(\eta)} q}. \tag{9.29}$$

Now, we put (9.23), (9.25), and (9.29) together, and get (for every $\beta > 0$)

$$A \vdash_8 \mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}} \langle h_{\boldsymbol{u}}^{(a)}, \mathrm{T}_{1-\gamma} h_{\boldsymbol{u}}^{(a)} \rangle \le \frac{1}{q^{1+\Omega(\gamma)}} + \frac{1}{4q^{\cdot 3}} \left(\beta \cdot q^{\cdot 1\log q} + \left(1 + \frac{1}{\beta}\right) \cdot \frac{\log q}{N^{\Omega(\eta)} q} \right). \tag{9.30}$$

We establish (9.19) by choosing $\beta = N^{c\cdot\eta}$ where $c > 0$ is a constant depending on the coefficient hidden in the $\Omega(\eta)$ notation in (9.30). $\square$

9.4.1 Proof of Claim 9.4.2

It remains to prove Claim 9.4.2.

Proof of Claim 9.4.2. By fact Fact 8.3.5, we know that

$$A \vdash_2 f_u^{(c)}(x) \leq 1,$$

and hence

$$A \vdash_2 f_u^{(c)}(x)^2 \leq 1,$$

for all $u \in \mathcal{V}, x \in \mathbb{Z}_q^N, c \in \mathbb{Z}_q$. Therefore, by Fact 8.3.15, we have

$$A \vdash_2 \tilde{f}_v^{(a)}(x)^2 \leq 1 \tag{9.31}$$

for all $x \in \mathbb{Z}_q^N$.

Since $\mathbf{Inf}_i^{\leq .02 \log q}(\tilde{f}_v^{(a)}) = \sum_{\alpha:\alpha_i > 0, |\alpha| \leq .02 \log q} \widehat{\tilde{f}_v^{(a)}}(\alpha)^2$, we have

$$A \vdash_2 \mathbf{Inf}_i^{\leq .02 \log q}(\tilde{f}_v^{(a)}) \leq \sum_{\alpha} \widehat{\tilde{f}_v^{(a)}}(\alpha)^2 = \mathop{\mathbf{E}}_{\boldsymbol{x}} \tilde{f}_v^{(a)}(\boldsymbol{x})^2.$$

Together with (9.31), we have

$$A \vdash_2 \mathbf{Inf}_i^{\leq .02 \log q}(\tilde{f}_v^{(a)}) \leq 1. \tag{9.32}$$

On the other hand, since $\mathbf{Inf}_i^{\leq .02 \log q}(\tilde{f}_v^{(a)})$ is a sum of squared linear forms, we can multiply $\mathbf{Inf}_i^{\leq .02 \log q}(\tilde{f}_v^{(a)})$ on the both sides of (9.32), and get

$$A \vdash_4 \mathbf{Inf}_i^{\leq .02 \log q}(\tilde{f}_v^{(a)})^2 \leq \mathbf{Inf}_i^{\leq .02 \log q}(\tilde{f}_v^{(a)}),$$

which is the desired statement. □

9.5 Refuting Instances based on Short Code

In this section, we consider the UNIQUEGAMES instances obtained by composing the KV UNIQUEGAMES integrality gap instances with the "short code" gadget reduction constructed in [35].

The following analog of Theorem 9.0.8 holds.

Theorem 9.5.1. *Let G be an n-variable* UNIQUEGAMES *instance with label-size q obtained by composing the KV "quotient noisy cube"* UNIQUEGAMES *integrality gap instances with the "short code" gadget reduction constructed in [35] so that the best assignment to G's variables satisfies at most an ϵ fraction of the constraints. When ϵ is sufficiently small and n is sufficiently large, there is a degree-8 SOS refutation for the statement that the best assignment to G's variables satisfy at least $1/100$ fraction of the constraints.*

The proof of Theorem 9.5.1 is almost literally the same as the proof of Theorem 9.0.8. In the following, we sketch the main arguments why the proof doesn't have to change. First, several of the results of the previous sections apply to general graphs and instances of UNIQUEGAMES. In particular, the proofs in Section 9.3 does not need to change. The proofs in 9.2 will still go through assuming the invariance principle result (in Section 9.1) for the type of graphs we are interested in; and the proofs in Section 9.4 apply to general gadget-composed instances of unique games assuming a "Majority Is Stablest" result for the gadget. In fact, the only parts that require further justification are the invariance principle (the proofs in Section 9.1) and hypercontractivity bound (Theorem 8.4.3). Both the invariance principle and the hypercontractivity bound are about the fourth moment of a low-degree Fourier polynomial (whose coefficients are fictitious random variables). For the construction of [35], we need to argue about the fourth moment with respect to a different distribution over inputs. (Instead of the uniform distribution, [35] considers a distribution over inputs related to the Reed–Muller code.) However, this distribution happens to be k-wise independent for $k/4$ larger than the degree of our Fourier polynomial. Hence, as a degree-4 polynomial in Fourier coefficients, the fourth moment with respect to the [35]-input distribution is the same as with respect to the uniform distribution, which considered here.

Chapter 10

SOS proofs of SSE in the Noisy Hypercube, KKL, and the analysis of the DKSV BALANCEDSEPARATOR instances

In this chapter, we given an SOS proof for the KKL theorem, and use this to show that degree-4 SOS proofs certify the lower bound for the optimal value of the DKSV BALANCEDSEPARATOR instances up to only a constant factor.

10.1 An SOS proof of small-set expansion in the noisy hypercube

The following well-known theorem concerning small-set expansion (SSE) in the hypercube is due to Kahn, Kalai, and Linial [129]:

Noisy Hypercube SSE Theorem. *Let* $f : \{-1, 1\}^n \to \{-1, 0, 1\}$. *Then for any* $0 \leq \rho \leq 1$,

$$\mathbf{Stab}_\rho[f] \leq \mathbf{E}[f^2]^{2/(1+\rho)},$$

where $\mathbf{Stab}_\rho[f]$ *denotes* $\langle f, \mathrm{T}_\rho f\rangle = \|\mathrm{T}_{\sqrt{\rho}} f\|_2^2$.

Proof.

$$\mathbf{Stab}_\rho[f] = \|\mathrm{T}_{\sqrt{\rho}} f\|_2^2 \leq \|f\|_{1+\rho}^2 = \mathbf{E}[|f|^{1+\rho}]^{2/(1+\rho)} = \mathbf{E}[f^2]^{2/(1+\rho)},$$

where the inequality is hypercontractivity (the Hölder dual of Theorem 8.4.1). $\square$

We remark on two special cases:

$$\mathbf{Stab}_{\frac{1}{3}}[f] \le \mathbf{E}[f^2]^{3/2}, \qquad \mathbf{Stab}_{\frac{1}{\sqrt{3}}}[f] \le \mathbf{E}[f^2]^{3-\sqrt{3}} \le \mathbf{E}[f^2]^{1.2679}.$$

We do not know how to obtain a low-degree SOS proof of either inequality. Nevertheless, we come close in the following theorem. We remark that its proof is very similar to the one in Section 9.3.

Theorem 10.1.1. *(SOS proof of a weakened special case of the Noisy Hypercube SSE Theorem.)*

Let $n \in \mathbb{N}$, and for each $x \in \{-1,1\}^n$ let $f(x)$ be an indeterminate. Then for any real $\epsilon > 0$,

$$\{f(x) = f(x)^3 : \forall x\} \quad \vdash_4 \quad \mathbf{Stab}_{\frac{1}{\sqrt{3}}}[f] \le \mathop{\mathbf{E}}_{\boldsymbol{x}}[f(\boldsymbol{x})^2]\left(\tfrac{3\epsilon}{4} + \tfrac{1}{4\epsilon^3}\,\mathbf{E}[f(\boldsymbol{x})^2]\right).$$

Remark 10.1.1. *From this we can deduce that if $f : \{-1,1\}^n \to \{-1,0,1\}$ is an ordinary function then $\mathbf{Stab}_{\frac{1}{\sqrt{3}}}[f] \le \mathbf{E}[f(\boldsymbol{x})^2]^{5/4}$, by taking $\epsilon = \mathbf{E}[f(\boldsymbol{x})^2]^{1/4}$.*

Proof. From Fact 8.3.12 (and the trivial fact $Y = Y^3 \vdash_4 Yp^2 = Y^4$) we may easily deduce

$$Y = Y^3 \quad \vdash_4 \quad YZ \le \tfrac{3\epsilon}{4}Y^2 + \tfrac{1}{4\epsilon^3}Z^4.$$

Since $\mathbf{Stab}_{\frac{1}{\sqrt{3}}}[f] = \mathbf{E}_{\boldsymbol{x}}[f(\boldsymbol{x})\mathrm{T}_{\frac{1}{\sqrt{3}}}f(\boldsymbol{x})]$ we may therefore obtain

$$\{f(x) = f(x)^3 : \forall x\} \quad \vdash_4 \quad \mathbf{Stab}_{\frac{1}{\sqrt{3}}}[f] \le \tfrac{3\epsilon}{4}\mathop{\mathbf{E}}_{\boldsymbol{x}}[f(\boldsymbol{x})^2] + \tfrac{1}{4\epsilon^3}\mathop{\mathbf{E}}_{\boldsymbol{x}}[\mathrm{T}_{\frac{1}{\sqrt{3}}}f(\boldsymbol{x})^4].$$

The result now follows from Theorem 8.4.4. $\square$

10.2 The KKL Theorem

With the Noisy Hypercube SSE Theorem in hand, we can now give an SOS proof of the famed KKL Theorem [129], the key ingredient in the analysis of the DKSV BALANCED-SEPARATOR instances.

Theorem 10.2.1. *(SOS proof of the KKL Theorem.)*

Let $n \in \mathbb{N}$, and for each $x \in \{-1,1\}^n$ let $f(x)$ be an indeterminate. Let τ be an indeterminate. Then for any reals $\epsilon > 0$, $K \geq 2$,

$$\{f(x)^2 = 1 : \forall x\} \cup \{\mathbf{Inf}_i[f] \leq \tau : \forall i \in [n]\} \quad \vdash_4 \quad \mathbf{Var}[f] \leq \left(\frac{\sqrt{3}^{K-1}}{K}\left(\tfrac{3\epsilon}{4} + \tfrac{\tau}{4\epsilon^3}\right) + \frac{1}{K}\right)\mathbf{I}[f].$$

Remark 10.2.1. *From this we can deduce that if $f : \{-1,1\}^n \to \{-1,1\}$ is an ordinary function and $\tau \leq \frac{1}{9}$ is a positive real such that $\mathbf{Inf}_i[f] \leq \tau$ for all i, then $\mathbf{I}[f] \geq \frac{1}{2}\log_9(\frac{9}{\tau}) \cdot \mathbf{Var}[f]$. This follows by taking $\epsilon = \tau^{1/4}$ and $K = \log_9(\frac{9}{\tau})$.*

Proof. We may apply Theorem 10.1.1 to each of the derivative "functions"

$$\mathrm{D}_i f(x) = \frac{f(x^{(i\mapsto 1)}) - f(x^{(i\mapsto -1)})}{2}.$$

(These are actually sets of indeterminates, each of which is a homogeneous linear form in the indeterminates $f(x)$.) We can obtain the hypothesis $\mathrm{D}_i f(x) = \mathrm{D}_i f(x)^3$ from the hypotheses $f(x)^2 = 1$ via Fact 8.3.6. We deduce

$$\begin{aligned}\{f(x)^2 = 1 : \forall x\} \quad \vdash_4 \quad & \mathbf{Stab}_{\frac{1}{\sqrt{3}}}[\mathrm{D}_i f] \leq \mathop{\mathbf{E}}_{\boldsymbol{x}}[\mathrm{D}_i f(\boldsymbol{x})^2]\left(\tfrac{3\epsilon}{4} + \tfrac{1}{4\epsilon^3}\mathbf{E}[\mathrm{D}_i f(\boldsymbol{x})^2]\right) \\ \Leftrightarrow \quad & \sum_{S \ni i} (\tfrac{1}{\sqrt{3}})^{|S|-1}\widehat{f}(S)^2 \leq \mathbf{Inf}_i[f]\left(\tfrac{3\epsilon}{4} + \tfrac{1}{4\epsilon^3}\mathbf{Inf}_i[f]\right)\end{aligned}$$

for each $i \in [n]$. Further, since $\mathbf{Inf}_i[f]$ is SOS and of degree 2 we have

$$\mathbf{Inf}_i[f] \leq \tau \quad \vdash_4 \quad \mathbf{Inf}_i[f] \cdot (\tfrac{\tau}{4\epsilon^3} - \tfrac{1}{4\epsilon^3}\mathbf{Inf}_i[f]) \geq 0.$$

Adding the previous two deductions yields

$$\{f(x)^2 = 1 : \forall x\} \cup \{\mathbf{Inf}_i[f] \leq \tau : \forall i \in [n]\} \quad \vdash_4 \quad \sum_{S \ni i} (\tfrac{1}{\sqrt{3}})^{|S|-1}\widehat{f}(S)^2 \leq \mathbf{Inf}_i[f] \cdot \left(\tfrac{3\epsilon}{4} + \tfrac{\tau}{4\epsilon^3}\right)$$

for each i. Now adding over all $i \in [n]$ gives

$$\{f(x)^2 = 1 : \forall x\} \cup \{\mathbf{Inf}_i[f] \leq \tau : \forall i \in [n]\} \quad \vdash_4 \quad \sum_{S \subseteq [n]} |S|(\tfrac{1}{\sqrt{3}})^{|S|-1}\widehat{f}(S)^2 \leq \left(\tfrac{3\epsilon}{4} + \tfrac{\tau}{4\epsilon^3}\right)\mathbf{I}[f].$$

Moreover, since $s(\frac{1}{\sqrt{3}})^{s-1} \geq K(\frac{1}{\sqrt{3}})^{K-1} - s(\frac{1}{\sqrt{3}})^{K-1}$ holds for all $s \in [n]$ (consider $s \leq K$ and $s \geq K$), it follows that

$$\vdash_2 \quad \sum_{S \subseteq [n]} |S|(\tfrac{1}{\sqrt{3}})^{|S|-1}\widehat{f}(S)^2 \geq \tfrac{K}{\sqrt{3}^{K-1}}\mathbf{Var}[f] - \tfrac{1}{\sqrt{3}^{K-1}}\mathbf{I}[f].$$

By combining the previous two deductions and doing some rearranging, we obtain

$$\{f(x)^2 = 1 : \forall x\} \cup \{\mathbf{Inf}_i[f] \le \tau : \forall i \in [n]\} \quad \vdash_4 \quad \mathbf{Var}[f] \le \left(\frac{\sqrt{3}^{K-1}}{K}\left(\tfrac{3\epsilon}{4} + \tfrac{\tau}{4\epsilon^3}\right) + \tfrac{1}{K}\right)\mathbf{I}[f],$$

as claimed. □

We can now easily deduce (an SOS proof of) the fact that if $f : \{-1,1\}^n \to \{-1,1\}$ has constant variance and all its influences equal then its total influence is $\Omega(\log n)$. For the application to BALANCEDSEPARATOR, we will in fact need a slightly more technical statement:

Corollary 10.2.2. *(SOS proof of KKL for equal-influence functions.)*

Let $n \ge 81$ be an integer and for each $x \in \{-1,1\}^n$ let $f(x)$ be an indeterminate. Define

$$\begin{aligned} A = \{f(x)^2 = 1 : \forall x\} \cup \{\mathbf{Inf}_i[f] \le \tau : \forall i \in [n]\} \\ \cup \{\mathbf{Inf}_i[f] = \mathbf{Inf}_j[f] : \forall i,j \in [n]\} \cup \{\mathbf{Var}[f] \ge \tfrac{3}{4}\} \cup \{\mathbf{I}[f] \le \tfrac{1}{20}\ln n\}. \end{aligned}$$

Then $A \vdash_4 -1 \ge 0$.

In fact, the result holds even if we change the equal-influences assumption $\{\mathbf{Inf}_i[f] = \mathbf{Inf}_j[f] : \forall i,j \in [n]\}$ to the weaker pair of assumptions $\{\mathbf{Inf}_i[f] = \mathbf{Inf}_j[f] : \forall i,j \le n/2\}$ and $\{\mathbf{Inf}_{i'}[f] = \mathbf{Inf}_{j'}[f] : \forall i',j' > n/2\}$ (assume n even).

Proof. We will prove the "in fact" statement, assuming n is even. (The reader will see why the original statement is also true when n is odd.) Define $\mathbf{I}^{(1)}[f] = \sum_{i \le n/2} \mathbf{Inf}_i[f]$ and $\mathbf{I}^{(2)}[f] = \sum_{i > n/2} \mathbf{Inf}_i[f]$, so $\mathbf{I}[f] = \mathbf{I}^{(1)}[f] + \mathbf{I}^{(2)}[f]$. Note that

$$\{\mathbf{Inf}_i[f] = \mathbf{Inf}_j[f] : \forall i,j \le n/2\} \quad \vdash_2 \quad \mathbf{Inf}_i[f] = \tfrac{2}{n}\mathbf{I}^{(1)}[f]$$

for each $i \le n/2$, and similarly for $i > n/2$. Since $\mathbf{I}^{(1)}[f], \mathbf{I}^{(2)}[f]$ are themselves SOS and of degree 2, we get

$$\{\mathbf{Inf}_i[f] = \mathbf{Inf}_j[f] : \forall i,j \le n/2 \,\&\, \forall i,j > n/2\} \quad \vdash_2 \quad \mathbf{Inf}_i[f] \le \tfrac{2}{n}\mathbf{I}^{(1)}[f] + \tfrac{2}{n}\mathbf{I}^{(2)}[f] = \tfrac{2}{n}\mathbf{I}[f]$$

for each $i \in [n]$. (Note that with the basic equal-influences assumption we can obtain the even stronger conclusion $\mathbf{Inf}_i[f] = \frac{1}{n}\mathbf{I}[f]$ for each $i \in [n]$.) We can now employ Theorem 10.2.1, replacing τ by $\frac{2}{n}\mathbf{I}[f]$. Using also $\mathbf{Var}[f] \ge \frac{3}{4}$, we obtain that for any reals $\epsilon > 0$, $K \ge 2$,

$$A \setminus \{\mathbf{I}[f] \le \tfrac{1}{20}\ln n\} \quad \vdash_4 \quad \tfrac{3}{4} \le \left(\frac{\sqrt{3}^{K-1}}{K}\left(\tfrac{3\epsilon}{4} + \tfrac{1}{2\epsilon^3 n}\mathbf{I}[f]\right) + \tfrac{1}{K}\right)\mathbf{I}[f].$$

Select $K = \log_9(9n^{1/2})$ and $\epsilon = n^{-1/8}$ to obtain

$$\begin{aligned} A \setminus \{\mathbf{I}[f] \le \tfrac{1}{20}\ln n\} \quad \vdash_4 \quad \tfrac{3}{4} &\le \left(\tfrac{n^{1/8}}{\log_9(9n^{1/2})}\left(\tfrac{3}{4}n^{-1/8} + \tfrac{1}{2}n^{-5/8}\mathbf{I}[f]\right) + \tfrac{1}{\log_9(9n^{1/2})}\right)\mathbf{I}[f] \\ &= \tfrac{7}{2\log_9(81n)}\mathbf{I}[f] + \tfrac{1}{n^{1/2}\log_9(81n)}\mathbf{I}[f]^2. \end{aligned} \tag{10.1}$$

We now employ $\mathbf{I}[f] \le \frac{1}{20}\ln n$. Since $\mathbf{I}[f]$ is SOS and of degree 2 we also have

$$\mathbf{I}[f] \le \tfrac{1}{20}\ln n \quad \vdash_4 \quad \mathbf{I}[f]^2 \le \tfrac{1}{400}\ln^2 n.$$

Substituting this into (10.1) yields

$$A \quad \vdash_4 \quad \tfrac{3}{4} \le \tfrac{7}{2\log_9(81n)} \cdot \tfrac{1}{20}\ln n + \tfrac{1}{n^{1/2}\log_9(81n)} \cdot \tfrac{1}{400}\ln^2 n \le \tfrac{7}{20}\ln(3) \le 0.4,$$

whence $A \vdash_4 -1 \ge 0$. $\square$

10.3 Analysis of the DKSV BALANCEDSEPARATOR instances

We recall the BALANCEDSEPARATOR problem: Given is an undirected multigraph $G = (V, E)$. It is required to find a cut $S \subseteq V$ with $\frac{1}{3} \le \frac{|S|}{|V|} \le \frac{2}{3}$ so as to minimize $\frac{E(S,\overline{S})}{|E|}$. The natural polynomial optimization formulation has an indeterminate $f(x)$ for each vertex $x \in V$:

$$\begin{aligned} \min \quad & \tfrac{1}{|E|}\sum_{(x,y)\in E}\left(\tfrac{f(x)-f(y)}{2}\right)^2 \\ \text{s.t.} \quad & f(x)^2 = 1 \quad \forall x \in V, \\ & \left(\tfrac{1}{|V|}\sum_{x\in V} f(x)\right)^2 \le \tfrac{1}{9}. \end{aligned}$$

Thus as discussed in Section 8.2, the degree-4 SOS SDP hierarchy will use binary search to compute the largest β for which

$$\{f(x)^2 = 1 : \forall x \in V\} \cup \left\{\left(\tfrac{1}{|V|}\sum_{x\in V} f(x)\right)^2 \le \tfrac{1}{9}\right\} \cup \left\{\tfrac{1}{|E|}\sum_{(x,y)\in E}\left(\tfrac{f(x)-f(y)}{2}\right)^2 \le \beta\right\} \quad \vdash_4 \quad -1 \ge 0.$$

The DKSV instances. We now recall the DKSV BALANCEDSEPARATOR instances [79]. The instances $G = (V, E) = (V_N, E_N)$ are parameterized by primes N. Let $\mathcal{F} = \{-1,1\}^N \times \{-1,1\}^N$, thought of as the $2N$-dimensional hypercube graph. Let σ act on elements $(x, y) \in \mathcal{F}$ by cyclic rotation of both halves: $\sigma(x, y) = (x_N, x_1, \ldots, x_{N-1}, y_N, y_1, \ldots, y_{N-1})$. The elements $\sigma, \sigma^2, \ldots, \sigma^N = id$ form a group acting on $\mathcal{F}$, partitioning it into orbits $\mathcal{O}_1, \ldots, \mathcal{O}_m$; 4 of these orbits have cardinality 1 and the remaining $(2^{2N} - 4)/N$ have cardinality N. A cardinality-N orbit $\mathcal{O}$ is called "nearly orthogonal" if for all distinct $(x, y), (x', y') \in \mathcal{O}$ it holds that $|\langle (x, y), (x', y') \rangle| \leq 8\sqrt{N \log N}$. Presuming that N is sufficiently large, [79] shows that the number n of nearly orthogonal orbits satisfies $(1 - 4/N^2)m \leq n \leq m$. (This implies $N = \Theta(\log n)$.) For typographic simplicity the nearly orthogonal orbits are assumed to be $\{\mathcal{O}_1, \ldots, \mathcal{O}_n\}$, and this set is taken to be the vertex set V. We write $L \subseteq \mathcal{F}$ for the "leftover" elements contained in orbits $\mathcal{O}_{n+1}, \ldots, \mathcal{O}_m$; writing $\epsilon = \frac{|L|}{2^{2N}}$ we have $\epsilon = O(1/N^2)$. The edges E in G are given by the usual hypercube edges in $\mathcal{F}$. More precisely, any pair $\mathcal{O}, \mathcal{O}' \in V$ have either N or 0 edges between them, according to whether or not there exist $(x, y) \in \mathcal{O}, (x', y') \in \mathcal{O}'$ at Hamming distance 1 in $\mathcal{F}$. There are no self-loops in G because of the near orthogonality property. The set of edges E is in one-to-one correspondence with a subset of (almost all the) hypercube edges in $\mathcal{F}$; specifically, all those not incident on L. The authors of [79] use the KKL Theorem to prove:

Theorem 10.3.1. *The DKSV* BALANCEDSEPARATOR *instances have optimum value* $\Omega(\frac{\log\log n}{\log n})$.

(Although we haven't formally verified it, it's very likely that the optimum value of these instances is also $O(\frac{\log\log n}{\log n})$, at least for infinitely many N. The reason is that there is a σ-invariant function $f : \mathcal{F} \to \{-1, 1\}$ of constant variance and total influence $\Omega(\log N)$; namely, $f(x, y) = 1$ if $x \in \{-1, 1\}^N$ contains a "run" (with wraparound) of length $\lfloor \log_2 N - \log_2 \log\log N \rfloor$.)

On the other hand, the main result of [79] is the following:

Theorem 10.3.2. *The standard SDP relaxation with triangle inequalities for the DKSV* BALANCEDSEPARATOR *instances has value* $O(\frac{1}{\log n})$.

We show here that this factor $\Theta(\log\log n)$ gap is eliminated when the degree-4 SOS relaxation is used.

Theorem 10.3.3. *The degree-*4 *SOS relaxation for the DKSV* BALANCEDSEPARATOR *instances has value* $\Omega(\frac{\log\log n}{\log n})$.

Proof. We need to show

$$\{f(\mathcal{O})^2 = 1 : \forall \mathcal{O} \in V\} \cup \left\{ \left(\tfrac{1}{n} \sum_{\mathcal{O} \in V} f(\mathcal{O}) \right)^2 \leq \tfrac{1}{9},\ \tfrac{1}{|E|} \sum_{(\mathcal{O},\mathcal{O}') \in E} \left(\tfrac{f(\mathcal{O}) - f(\mathcal{O}')}{2} \right)^2 \leq c \tfrac{\log \log n}{\log n} \right\}$$
$$\vdash_4 -1 \geq 0 \quad (10.2)$$

for some constant $c > 0$ (and N sufficiently large).

Introduce indeterminates $g(x)$ for all $x \in \mathcal{F} = \{-1,1\}^N \times \{-1,1\}^N$. By Corollary 10.2.2 it is possible to write

$$\begin{aligned} -1 &= u_0 + \sum_{x \in \mathcal{F}} v_x (g(x)^2 - 1) \\ &+ \sum_{\substack{1 \leq i < j \leq N \\ N+1 \leq i < j \leq 2N}} w_{ij} (\mathbf{Inf}_i[g] - \mathbf{Inf}_j[g]) + u_1(\mathbf{Var}[g] - \tfrac{3}{4}) + u_2(\tfrac{1}{20} \ln(2N) - \mathbf{I}[g]), \quad (10.3) \end{aligned}$$

where u_0, u_1, u_2 are SOS (in the variables $g(x)$) and all summands have degree at most 4. Now substitute into this identity $g(x) = f(\mathcal{O})$ for each $x \in \mathcal{O} \in V$, and also substitute $g(x) = 1$ for each $x \in \mathcal{F}$ which is not contained in any $\mathcal{O} \in V$. We now consider what happens to each term in (10.3).

First, we notice that the degree of each term cannot increase. The polynomial u_0 (now over indeterminates $f(\mathcal{O})$) remains SOS. The next term, $\sum_{x \in \mathcal{F}} v_x (g(x)^2 - 1)$, becomes of the form $\sum_{\mathcal{O} \in V} v'_{\mathcal{O}} (f(\mathcal{O})^2 - 1)$ for some polynomials $v'_{\mathcal{O}}$. We claim that each summand $w_{ij}(\mathbf{Inf}_i[g] - \mathbf{Inf}_j[g])$ in the next term drops out entirely. This is because when g is viewed as mapping from $\mathcal{F}$ to the set of homogeneous degree-1 polynomials in the $f(\mathcal{O})$'s, it is invariant under the action of σ, by construction. From this it follows that $\mathbf{Inf}_i[g] = \mathbf{Inf}_j[g]$ formally as polynomials for all $1 \leq i < j \leq N$ and $N+1 \leq i < j \leq 2N$.

Next we come to the term $u_1(\mathbf{Var}[g] - \frac{3}{4})$. We have

$$\mathbf{Var}[g] - \tfrac{3}{4} = \mathop{\mathbf{E}}_{\boldsymbol{x} \in \mathcal{F}} [g(\boldsymbol{x})^2] - \tfrac{3}{4} - \mathop{\mathbf{E}}_{\boldsymbol{x} \in \mathcal{F}} [g(\boldsymbol{x})]^2.$$

Even after our substitution, $\mathbf{E}_{\boldsymbol{x} \in \mathcal{F}}[g(\boldsymbol{x})^2] - \frac{3}{4}$ will provably equal $\frac{1}{4}$ under the assumption $\{f(\mathcal{O})^2 = 1 : \forall \mathcal{O} \in V\}$, so it remains to focus on

$$\mathop{\mathbf{E}}_{\boldsymbol{x} \in \mathcal{F}} [g(\boldsymbol{x})]^2 = \left(\epsilon + (1 - \epsilon) \tfrac{1}{n} \sum_{\mathcal{O} \in V} f(\mathcal{O}) \right)^2.$$

Recalling that $\vdash_2 (Y + Z)^2 \leq 2Y^2 + 2Z^2$, we deduce

$$\left(\tfrac{1}{n} \sum_{\mathcal{O} \in V} f(\mathcal{O}) \right)^2 \leq \tfrac{1}{9} \quad \vdash_2 \quad \left(\epsilon + (1 - \epsilon) \tfrac{1}{n} \sum_{\mathcal{O} \in V} f(\mathcal{O}) \right)^2 \leq 2\epsilon^2 + 2(1 - \epsilon)^2 \cdot \tfrac{1}{9} \leq \tfrac{1}{4}$$

(for N sufficiently large, since $\epsilon = O(1/N^2)$), as needed.

Finally we come to the term $u_2(\frac{1}{20}\ln(2N) - \mathbf{I}[g])$. Let ϵ' denote the fraction of hypercube edges in $\mathcal{F}$ which are incident on L; note that $\epsilon' \leq 2\epsilon = O(1/N^2)$. After our substitution, we have

$$\begin{aligned}\tfrac{1}{20}\ln(2N) - \mathbf{I}[g] &= \tfrac{1}{20}\ln(2N) - (2N) \mathop{\mathbf{E}}_{\substack{\text{edge } (\boldsymbol{x},\boldsymbol{y})\\ \text{in } \mathcal{F}}}\left[\left(\tfrac{g(\boldsymbol{x})-g(\boldsymbol{y})}{2}\right)^2\right] \\ &= \tfrac{1}{20}\ln(2N) - (2N)(1-\epsilon') \cdot \tfrac{1}{|E|}\sum_{(\mathcal{O},\mathcal{O}')\in E}\left(\tfrac{f(\mathcal{O})-f(\mathcal{O}')}{2}\right)^2 - \epsilon' \cdot (*),\end{aligned} \tag{10.4}$$

where $(*)$ is the average of a number of terms, some of which are $(\frac{1-1}{2})^2 = 0$ and some of which are of the form

$$\left(\tfrac{f(\mathcal{O}_i)-1}{2}\right)^2 = 1 + \tfrac{1}{2}(f(\mathcal{O}_i)^2 - 1) - \tfrac{1}{4}(f(\mathcal{O}_i)+1)^2.$$

The above shows that $\{f(\mathcal{O})^2 = 1 : \forall \mathcal{O} \in V\} \vdash_2 \left(\frac{f(\mathcal{O}_i)-1}{2}\right)^2 \leq 1$. Hence

$$\begin{aligned}&\{f(\mathcal{O})^2 = 1 : \forall \mathcal{O} \in V\} \cup \left\{\tfrac{1}{|E|}\sum_{(\mathcal{O},\mathcal{O}')\in E}\left(\tfrac{f(\mathcal{O})-f(\mathcal{O}')}{2}\right)^2 \leq c\tfrac{\log\log n}{\log n}\right\} \\ &\qquad\qquad \vdash_2 \quad (10.4) \geq \tfrac{1}{20}\ln(2N) - (2N)(1-\epsilon')\cdot c\tfrac{\log\log n}{\log n} - \epsilon',\end{aligned}$$

which is nonnegative for c sufficiently small, since $N = \Theta(\log n)$ and $\epsilon' = O(1/N^2)$. Thus we have verified (10.2). $\square$

Chapter 11

SOS proofs of the CLT, the $\frac{2}{\pi}$ Theorem, and the analysis of the KV MAXCUT instances

In this chapter, we give an SOS proof for the $\frac{2}{\pi}$ theorem, and use this to show that constant-degree SOS proofs certify a better upper bound on the optimal value of the KV MAXCUT instances (than the upper bound given by the Goemans-Williamson algorithm).

11.1 An invariance theorem for polynomials of linear forms

Theorem 11.1.1. *(SOS proof of an Invariance Theorem for polynomials of linear forms.)*

Let $a_1, \dots, a_n$ be indeterminates. For any real vector $z = (z_1, \dots, z_n)$, let $\ell(z)$ denote the homogeneous linear polynomial $\ell(z) = a_1 z_1 + \cdots + a_n z_n$. Then for any even integer $k \geq 4$ we have

$$a_1^2 + \cdots + a_n^2 \leq 1 \quad \vdash_{2k} \quad \mathop{\mathbf{E}}_{\boldsymbol{G}}[\ell(\boldsymbol{G})^k] - k^{O(k)} \sum_{i=1}^{n} a_i^4 \leq \mathop{\mathbf{E}}_{\boldsymbol{x}}[\ell(\boldsymbol{x})^k] \leq \mathop{\mathbf{E}}_{\boldsymbol{G}}[\ell(\boldsymbol{G})^k],$$

where $\boldsymbol{G} = (\boldsymbol{G}_1, \dots, \boldsymbol{G}_n) \sim \mathrm{N}(0,1)^n$ and $\boldsymbol{x} \sim \{-1,1\}^n$ is uniform.

Remark 11.1.1. *It is easy to see that $\mathbf{E}_{\boldsymbol{x}}[\ell(\boldsymbol{x})^k] = \mathbf{E}_{\boldsymbol{G}}[\ell(\boldsymbol{G})^k]$ formally as polynomials for $k = 0, 1, 2, 3$, and any odd integer $k > 3$.*

Proof. For each integer $0 \le i \le n$, define the polynomial

$$P_i = \mathbf{E}[\ell(\boldsymbol{G}_1, \cdots, \boldsymbol{G}_i, \boldsymbol{x}_{i+1}, \cdots, \boldsymbol{x}_n)^k].$$

We will show for each $1 \le i \le n$ that

$$a_1^2 + \cdots + a_n^2 \le 1 \quad \vdash_{2k} \quad P_i - k^{O(k)} a_i^4 \le P_{i-1} \le P_i. \tag{11.1}$$

The desired result then follows by summing over i. So fix $1 \le i \le n$ and write $\ell(z) = \ell'(z') + a_i z_i$, where $z' = (z_1, \ldots, z_{i-1}, z_{i+1}, z_n)$ and

$$\ell'(z') = a_1 z_1 + \cdots + a_{i-1} z_{i-1} + a_{i+1} z_{i+1} + \cdots + a_n z_n$$

does not depend on the indeterminate a_i. Denoting $\boldsymbol{Z}' = (\boldsymbol{G}_1, \ldots, \boldsymbol{G}_{i-1}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_n)$ we have

$$\begin{aligned} P_i - P_{i-1} &= \mathop{\mathbf{E}}_{\boldsymbol{Z}'} \left[\mathop{\mathbf{E}}_{\boldsymbol{G}_i, \boldsymbol{x}_i} [(\ell'(\boldsymbol{Z}') + a_i \boldsymbol{G}_i)^k - (\ell'(\boldsymbol{Z}') + a_i \boldsymbol{x}_i)^k] \right] \\ &= \sum_{j=2}^{k/2} \binom{k}{2j} \left((2j-1)!! - 1\right) a_i^{2j} \mathop{\mathbf{E}}_{\boldsymbol{Z}'}[\ell'(\boldsymbol{Z}')^{k-2j}] \end{aligned} \tag{11.2}$$

where we used $\mathbf{E}[\boldsymbol{G}_i^r] = \mathbf{E}[\boldsymbol{x}_i^r] = 0$ for r odd and $\mathbf{E}[\boldsymbol{G}_i^r] = (r-1)!!$, $\mathbf{E}[\boldsymbol{x}_i^r] = 1$, for r even. The above polynomial is evidently SOS, justifying the second inequality in (11.1). As for the first inequality in (11.1), we have

$$a_1^2 + \cdots + a_n^2 \le 1 \quad \vdash_{2j} \quad a_i^{2j} \le a_i^4 \tag{11.3}$$

for each $i \in [n]$ and $2 \le j \le k/2$ because

$$\begin{aligned} a_i^4 - a_i^{2j} &= (1 - a_i^2)(a_i^4 + a_i^6 + a_i^8 + \cdots + a_i^{2j-2}) \\ &= \left((1 - \sum_{i'=1}^{n} a_{i'}^2) + \sum_{i' \ne i} a_{i'}^2 \right) (a_i^4 + a_i^6 + a_i^8 + \cdots + a_i^{2j-2}); \end{aligned}$$

and, we have

$$\begin{aligned} a_1^2 + \cdots + a_n^2 \le 1 \quad \vdash_{2(k-2j)} \quad \mathop{\mathbf{E}}_{\boldsymbol{Z}'}[\ell'(\boldsymbol{Z}')^{k-2j}] &\le k^{O(k)} \mathop{\mathbf{E}}_{\boldsymbol{Z}'}[\ell'(\boldsymbol{Z}')^2]^{k/2-j} \\ &= k^{O(k)}(a_1^2 + \cdots + a_n^2)^{k/2-j} \le k^{O(k)} \end{aligned} \tag{11.4}$$

by Corollary 8.4.7, the second inequality's SOS proof being

$$1 - (\sum a_{i'}^2)^{k/2-j} = 1 + (\sum a_{i'}^2) + (\sum a_{i'}^2)^2 + (\sum a_{i'}^2)^3 + \cdots + (\sum a_{i'}^2)^{k/2-j-1}.$$

Combining (11.3) and (11.4) via Fact 8.2.4

$$a_1^2 + \cdots + a_n^2 \le 1 \quad \vdash_{2k} \quad a_i^{2j} \mathop{\mathbf{E}}_{\boldsymbol{Z}'}[\ell'(\boldsymbol{Z}')^{k-2j}] \le k^{O(k)} a_i^4.$$

Using this in (11.2), along with $\binom{k}{2j}((2j-1)!!-1) \le k^{O(k)}$ for each j, yields the first inequality in (11.1), completing the proof. $\square$

11.2 An SOS proof of the $\frac{2}{\pi}$ Theorem

We require the below technical lemma giving a polynomial approximator to the absolute-value function. The proof uses some standard methods in approximation theory and is deferred to Section 11.4.

Lemma 11.2.1. *For any sufficiently small parameter $\delta > 0$, there exists a univariate, real, even polynomial $P(t) = Q(t^2)$ of degree at most $\widetilde{O}(1/\delta^2)$ such that:*

1. *$P(t) \ge |t|$ for all $t \in \mathbb{R}$;*
2. *$\mathbf{E}[P(\sigma \boldsymbol{g})] \le \sqrt{\frac{2}{\pi}} \cdot \sigma + \delta \le \left(\frac{1}{2}\sigma^2 + \frac{1}{\pi}\right) + \delta$ for all $0 \le \sigma \le 1$, where $\boldsymbol{g} \sim \mathrm{N}(0,1)$;*
3. *Each coefficient of P is at most $2^{O(d)}$ in absolute value.*

It is not hard to show that among degree-2 polynomials $P(t)$ with $P(t) \ge |t|$, the lowest possible value of $\mathbf{E}[P(\boldsymbol{g})]$ is 1, achieved by $P(t) = \frac{1}{2} + \frac{1}{2}t^2$. Interestingly, this is also the lowest possible value even when degree-4 is allowed:

Theorem 11.2.2. *Suppose $P(t)$ is a univariate real polynomial of degree at most 4 satisfying $P(t) \ge |t|$ for all real t. Then $\mathbf{E}_{\boldsymbol{g}\sim\mathrm{N}(0,1)}[P(\boldsymbol{g})] \ge 1$.*

Proof. Replacing $P(t)$ by $\frac{1}{2}(P(t) + P(-t))$ if necessary, we may assume $P(t)$ is even; i.e., $P(t) = a + bt^2 + ct^4$ for some real a, b, c. For any $M > 0$ we have

$$\begin{aligned}
& \tfrac{1}{M^2} \times (P(0) \ge 0) \;+\; \tfrac{M^2-3}{M^2-1} \times (P(1) \ge 1) \;+\; \tfrac{2}{(M^2-1)M^2} \times (P(M) \ge M) \\
\Rightarrow\quad & \tfrac{1}{M^2} \times (a \ge 0) \;+\; \tfrac{M^2-3}{M^2-1} \times (a+b+c \ge 1) \;+\; \tfrac{2}{(M^2-1)M^2} \times \left(a + M^2 b + M^4 c \ge M\right) \\
\Rightarrow\quad & a + b + 3c \ge 1 - \tfrac{2}{M(M+1)}.
\end{aligned}$$

This completes the proof because $\mathbf{E}[P(\boldsymbol{g})] = a+b+3c$ and M may be arbitrarily large. $\square$

Remark 11.2.1. *Once we allow degree* 6 *it is possible to obtain a bound strictly smaller than* 1*. For example,* $P(t) = .333 + .815t^2 - .136t^4 + .01t^6 \geq |t|$ *pointwise, and* $\mathbf{E}[P(\boldsymbol{g})] = .89$.

The following "$\frac{2}{\pi}$ Theorem", due to [141], is essentially the special case of the Majority Is Stablest Theorem in which $\rho \to 0^+$. We reproduce the proof.

Theorem 11.2.3. *Let* $f : \{-1,1\}^n \to [-1,1]$ *and assume* $|\widehat{f}(i)| \leq \epsilon$ *for all* $i \in [n]$*. Then* $\sum_{i=1}^n \widehat{f}(i)^2 \leq \frac{2}{\pi} + O(\epsilon)$.

Proof. Let $\ell : \{-1,1\}^n \to \mathbb{R}$ be $\ell(x) = \sum_{i=1}^n \widehat{f}(i)x_i$ and let $\sigma = \sqrt{\sum_{i=1}^n \widehat{f}(i)^2}$. Then

$$\sigma^2 = \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[f(\boldsymbol{x})\ell(\boldsymbol{x})] \leq \mathop{\mathbf{E}}_{\boldsymbol{x}}[|\ell(\boldsymbol{x})|] \leq \sigma \mathop{\mathbf{E}}_{\boldsymbol{g}\sim\mathrm{N}(0,1)}[|\boldsymbol{g}|] + O(\sigma\epsilon) = \sigma\left(\sqrt{\tfrac{2}{\pi}} + O(\epsilon)\right),$$

the inequality being Berry–Esseen. The result follows after dividing by σ and squaring. $\square$

Theorem 11.2.4. *(SOS proof of the Berry–Esseen Theorem with* ℓ_1 *functional.)*

Let $a_1, \dots, a_n$ *be indeterminates, and for each* $x \in \{-1,1\}^n$*, let* $f(x)$ *be an indeterminate. Let*

$$A = \{f(x) \geq -1, f(x) \leq 1 : \forall x\} \cup \{a_1^2 + \cdots + a_n^2 \leq 1\}.$$

Then for any small real $\delta > 0$*,*

$$A \quad \vdash_{\widetilde{O}(1/\delta^2)} \quad \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[f(\boldsymbol{x})(a_1\boldsymbol{x}_1 + \cdots + a_n\boldsymbol{x}_n)] \leq b + \delta + 2^{\widetilde{O}(1/\delta^2)} \sum_{i=1}^n a_i^4,$$

where we may choose either

$$b = \sqrt{\tfrac{2}{\pi}} \qquad \text{or} \qquad b = \tfrac{1}{2}(a_1^2 + \cdots + a_n^2) + \tfrac{1}{\pi}.$$

Proof. For each $x \in \{-1,1\}^n$, let $\ell(x)$ denote $a_1x_1 + \cdots + a_nx_n$, a homogeneous linear polynomial in the indeterminates a_i. Let $P(t) = Q(t^2) = \sum_{k=0,2,4,\dots,d} c_k t^k$ be the univariate real polynomial in Lemma 11.2.1, where $d = \deg(P) \leq \widetilde{O}(1/\delta^2)$. Since $P(t) \geq \pm t$ for all real t, Fact 8.3.1 tells us that $\vdash_d P(t) \geq \pm t$ in $\mathbb{R}[t]$. Using Fact 8.3.7 and substituting $t = \ell(x)$ we deduce

$$\{f(x) \geq -1, f(x) \leq 1\} \quad \vdash_{d+1} \quad f(x)\ell(x) \leq P(\ell(x)).$$

Averaging over x yields

$$\{f(x) \geq -1, f(x) \leq 1 : \forall x\} \quad \vdash_{d+1} \quad \mathop{\mathbf{E}}_{\boldsymbol{x}}[f(\boldsymbol{x})(a_1\boldsymbol{x}_1 + \cdots + a_n\boldsymbol{x}_n)] \leq \mathop{\mathbf{E}}_{\boldsymbol{x}}[P(\ell(\boldsymbol{x}))] = \sum_{k=0,2,4,\ldots,d} \mathop{\mathbf{E}}_{\boldsymbol{x}}[c_k\ell(\boldsymbol{x})^k]. \quad (11.5)$$

For each even $0 \leq k \leq d$, regardless of the sign of c_k, Theorem 11.1.1 implies that

$$a_1^2 + \cdots + a_n^2 \leq 1 \quad \vdash_{2d} \quad \mathop{\mathbf{E}}_{\boldsymbol{x}}[c_k\ell(\boldsymbol{x})^k] \leq \mathop{\mathbf{E}}_{\boldsymbol{G}}[c_k\ell(\boldsymbol{G})^k] + |c_k|k^{O(k)}\sum_{i=1}^n a_i^4.$$

Summing this over k, using $\sum_k |c_k|k^{O(k)} \leq d^{O(d)}$, and combining with (11.5) yields

$$A \quad \vdash_{2d} \quad \mathop{\mathbf{E}}_{\boldsymbol{x}}[f(\boldsymbol{x})(a_1\boldsymbol{x}_1 + \cdots + a_n\boldsymbol{x}_n)] \leq \mathop{\mathbf{E}}_{\boldsymbol{G}}[P(\ell(\boldsymbol{G}))] + d^{O(d)}\sum_{i=1}^n a_i^4. \quad (11.6)$$

Let σ^2 be shorthand for $\sum_{i=1}^n a_i^2$. Note that if we treat $a_1, \ldots, a_n$ as arbitrary *real numbers*, we have

$$\mathop{\mathbf{E}}_{\boldsymbol{G}}[P(\ell(\boldsymbol{G}))] = \mathop{\mathbf{E}}_{\boldsymbol{g}\sim\mathrm{N}(0,1)}[P(\sigma\boldsymbol{g})] = \mathop{\mathbf{E}}_{\boldsymbol{g}\sim\mathrm{N}(0,1)}[Q(\sigma^2\boldsymbol{g}^2)], \quad (11.7)$$

by the rotational symmetry of multivariate Gaussians. Since the left and right sides are polynomials in $a_1, \ldots, a_n$, it follows that (11.7) also holds as a formal polynomial identity over the indeterminates $a_1, \ldots, a_n$. Now temporarily view σ^2 as an indeterminate. From Lemma 11.2.1 we have that $\mathbf{E}_{\boldsymbol{g}\sim\mathrm{N}(0,1)}[Q(\sigma^2\boldsymbol{g}^2)]$ is upper-bounded by both $\sqrt{\frac{2}{\pi}} + \delta$ and $\frac{1}{2}\sigma^2 + \frac{1}{\pi} + \delta$ for all *real numbers* $0 \leq \sigma^2 \leq 1$. Thus from Fact 8.3.2 we have the following univariate SOS proof(s):

$$(1-\sigma^2)\sigma^2 \geq 0 \quad \vdash_{d/2} \quad \mathop{\mathbf{E}}_{\boldsymbol{g}\sim\mathrm{N}(0,1)}[Q(\sigma^2\boldsymbol{g}^2)] \leq \sqrt{\tfrac{2}{\pi}} + \delta,\ \tfrac{1}{2}\sigma^2 + \frac{1}{\pi} + \delta$$

(note that Q has even degree). Letting $\sigma^2 = \sum_{i=1}^n a_i^2$ again, we deduce that for either choice of b,

$$\begin{aligned} \left(1 - \sum_{i=1}^n a_i^2\right)\left(\sum_{i=1}^n a_i^2\right) \geq 0 \quad &\vdash_d \quad \mathop{\mathbf{E}}_{\boldsymbol{g}\sim\mathrm{N}(0,1)}[Q(\sigma^2\boldsymbol{g}^2)] \leq b + \delta \\ \Leftrightarrow \qquad a_1^2 + \cdots + a_n^2 \leq 1 \quad &\vdash_d \quad \mathop{\mathbf{E}}_{\boldsymbol{G}}[P(\ell(\boldsymbol{G}))] \leq b + \delta, \end{aligned}$$

using (11.7) and the fact that $\sum_{i=1}^n a_i^2$ is already SOS. Combining this with (11.6) yields

$$A \quad \vdash_{2d} \quad \mathop{\mathbf{E}}_{\boldsymbol{x}}[f(\boldsymbol{x})(a_1\boldsymbol{x}_1 + \cdots + a_n\boldsymbol{x}_n)] \le b + \delta + d^{O(d)} \sum_{i=1}^n a_i^4,$$

as needed. $\square$

Corollary 11.2.5. *(SOS proof of the $\frac{2}{\pi}$ Theorem.)*

For each $x \in \{-1,1\}^n$, let $f(x)$ be an indeterminate. Define $\widehat{f}(S)$ as usual and write $\widehat{f}(i) = \widehat{f}(\{i\})$ for short. Let

$$A = \{f(x) \ge -1, f(x) \le 1 : \forall x\}.$$

Then for each small real $\delta > 0$,

$$\begin{aligned} A \qquad & \vdash_{\widetilde{O}(1/\delta^2)} \qquad \sum_{i=1}^n \widehat{f}(i)^2 \le \tfrac{2}{\pi} + \delta + 2^{\widetilde{O}(1/\delta^2)} \sum_{i=1}^n \widehat{f}(i)^4, \\ A \cup \{\widehat{f}(i)^2 \le \tau : \forall i \in [n]\} \qquad & \vdash_{\widetilde{O}(1/\delta^2)} \qquad \sum_{i=1}^n \widehat{f}(i)^2 \le \tfrac{2}{\pi} + \delta + 2^{\widetilde{O}(1/\delta^2)} \cdot \tau. \end{aligned}$$

Proof. We wish to apply Theorem 11.2.4 with $a_i = \widehat{f}(i)$ for each $i \in [n]$. A standard proof shows that

$$\sum_{S \subseteq [n]} \widehat{f}(S)^2 = \mathop{\mathbf{E}}_{\boldsymbol{x} \sim \{-1,1\}^n}[f(\boldsymbol{x})^2]$$

and hence, using Fact 8.3.4,

$$\vdash_3 \quad \sum_{i=1}^n \widehat{f}(i)^2 \le 1. \tag{11.8}$$

We may therefore employ Theorem 11.2.4 (with $\delta/2$ instead of δ) to obtain

$$A \quad \vdash_{\widetilde{O}(1/\delta^2)}$$
$$\mathop{\mathbf{E}}_{\boldsymbol{x} \sim \{-1,1\}^n}[f(\boldsymbol{x})(\widehat{f}(1)\boldsymbol{x}_1 + \cdots + \widehat{f}(n)\boldsymbol{x}_n)] \le \tfrac{1}{2} \sum_{i=1}^n \widehat{f}(i)^2 + \tfrac{1}{\pi} + \tfrac{\delta}{2} + 2^{\widetilde{O}(1/\delta^2)} \sum_{i=1}^n \widehat{f}(i)^4.$$

But

$$\mathop{\mathbf{E}}_{\boldsymbol{x} \sim \{-1,1\}^n}[f(\boldsymbol{x})(\widehat{f}(1)\boldsymbol{x}_1 + \cdots + \widehat{f}(n)\boldsymbol{x}_n)] = \sum_{i=1}^n \widehat{f}(i)^2$$

is a polynomial identity so we deduce

$$\begin{aligned} A \quad \vdash_{\widetilde{O}(1/\delta^2)} \quad & \sum_{i=1}^{n} \widehat{f}(i)^2 \leq \tfrac{1}{2}\sum_{i=1}^{n} \widehat{f}(i)^2 + \tfrac{1}{\pi} + \tfrac{\delta}{2} + 2^{\widetilde{O}(1/\delta^2)} \sum_{i=1}^{n} \widehat{f}(i)^4 \\ \Leftrightarrow \quad & \sum_{i=1}^{n} \widehat{f}(i)^2 \leq \tfrac{2}{\pi} + \delta + 2^{\widetilde{O}(1/\delta^2)} \sum_{i=1}^{n} \widehat{f}(i)^4, \end{aligned}$$

completing the first part of the proof. Now adding the assumptions $\widehat{f}(i)^2 \leq \tau$ easily yields

$$A \cup \{\widehat{f}(i)^2 \leq \tau : \forall i \in [n]\} \quad \vdash_4 \quad \sum_{i=1}^{n} \widehat{f}(i)^4 \leq \tau \sum_{i=1}^{n} \widehat{f}(i)^2 \leq \tau$$

using (11.8) again. The proof is complete. $\square$

11.3 Analysis of the KV MAXCUT instances

We recall the MAXCUT problem: Given is an undirected weighted graph G on vertex set V in which the nonnegative edge weights sum to 1. We write $(\boldsymbol{x}, \boldsymbol{y}) \sim E$ to denote that $(\boldsymbol{x}, \boldsymbol{y})$ is a random edge chosen with probability equal to the edge weight. It is required to find a cut $S \subseteq V$ so as to maximize $\Pr_{(\boldsymbol{x},\boldsymbol{y})\sim E}[\boldsymbol{x} \in S, \boldsymbol{y} \notin S$ or vice versa$]$. The natural polynomial optimization formulation has an indeterminate $f(x)$ for each vertex $x \in V$:

$$\begin{aligned} \max \quad & \mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim E}[\tfrac{1}{2} - \tfrac{1}{2} f(\boldsymbol{x}) f(\boldsymbol{y})] \\ \text{s.t.} \quad & f(x)^2 = 1 \quad \forall x \in V. \end{aligned}$$

Thus as discussed in Section 8.2, the degree-d SOS SDP hierarchy will use binary search to compute the smallest β for which

$$\{f(x)^2 = 1 : \forall x \in V\} \cup \left\{ \mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim E}[\tfrac{1}{2} - \tfrac{1}{2} f(\boldsymbol{x}) f(\boldsymbol{y})] \geq \beta \right\} \quad \vdash_d \quad -1 \geq 0.$$

UNIQUEGAMES. The Khot–Vishoi (KV) instances of MAXCUT [144] are given by composing the KKMO "noise stability" reduction from [141] with the KV integrality gap instances for UNIQUEGAMES (UG). Our SOS proof of the $\frac{2}{\pi}$ Theorem gives us a "black-box" analysis of the KKMO reduction which can essentially be "plugged in" to a sufficiently strong SOS analysis of UG instances. Let us now recall the UNIQUEGAMES problem with label-size $k \in \mathbb{N}^+$. Given is a regular weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (self-loops allowed) with weights summing to 1. Also, given for each edge (u, v) is a permutation $\pi_{uv} : [k] \to [k]$. We write $(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\pi}) \sim \mathcal{E}$ to denote that edge $(\boldsymbol{u}, \boldsymbol{v})$ with permutation $\boldsymbol{\pi} = \pi_{\boldsymbol{uv}}$ is chosen with probability equal to its edge weight. The goal is to give a labeling $F : \mathcal{V} \to [k]$ so as to maximize $\Pr_{(\boldsymbol{u},\boldsymbol{v},\boldsymbol{\pi})\sim\mathcal{E}}[\boldsymbol{\pi}(F(\boldsymbol{u})) = F(\boldsymbol{v})]$. The natural polynomial optimization formulation has an indeterminate $X_{u,i}$ for each $u \in \mathcal{V}, i \in [k]$:

$$\begin{aligned}
\max \quad & \mathop{\mathbf{E}}_{(\boldsymbol{u},\boldsymbol{v},\boldsymbol{\pi})\sim\mathcal{E}}\Big[\sum_{i=1}^{k} X_{\boldsymbol{u},i} X_{\boldsymbol{v},\boldsymbol{\pi}(i)}\Big] = \mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}}\Big[\sum_{i=1}^{k} X_{\boldsymbol{u},i} \cdot \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim\boldsymbol{u}}[X_{\boldsymbol{v},\boldsymbol{\pi}(i)}]\Big] \\
\text{s.t.} \quad & X_{u,i}^2 = X_{u,i} \quad \forall u \in \mathcal{V}, i \in [k] \\
& \sum_{i=1}^{k} X_{u,i} = 1 \quad \forall u \in \mathcal{V},
\end{aligned}$$

where we write $(\boldsymbol{v}, \boldsymbol{\pi}) \sim u$ in place of $(u, \boldsymbol{v}, \boldsymbol{\pi}) \sim \mathcal{E}|_{\boldsymbol{u}=u}$ for brevity. Thus the degree-d SOS SDP hierarchy will use binary search to compute the smallest β for which

$$\{X_{u,i}^2 = X_{u,i} : \forall u \in \mathcal{V}, i \in [k]\} \cup \{\sum_{i=1}^{k} X_{u,i} = 1 : \forall u \in \mathcal{V}\} \cup \Big\{\mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}}\Big[\sum_{i=1}^{k} X_{\boldsymbol{u},i} \cdot \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim\boldsymbol{u}}[X_{\boldsymbol{v},\boldsymbol{\pi}(i)}]\Big] \geq \beta\Big\} \quad \vdash_d \quad -1 \geq 0.$$

In Section 9.3, we have shown that the degree-4 moment SDP proves that the KV family of UG instances has a very low optimum value. In fact we have shown something stronger; one only needs the hypotheses $X_{u,i}^2 \leq X_{u,i}$ and $(\operatorname{avg}_{u,i} X_{u,i})^2 \leq 1/k^2$. Let us make a somewhat more general definition which applies to SOS-refutations of any UG instances:

Definition 11.3.1. *Given a UG instance $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with label-size k, we say there is a* degree-d SOS refutation that the *fractional assignment* optimum is at least β *if*

$$\{X_{u,i} \geq 0 : \forall u \in \mathcal{V}, i \in [k]\} \cup \{\sum_{i=1}^{k} X_{u,i} \leq 1 : \forall u \in \mathcal{V}\} \cup \Big\{\mathop{\mathbf{E}}_{\boldsymbol{u}\in\mathcal{V}}\Big[\sum_{i=1}^{k} X_{\boldsymbol{u},i} \cdot \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim\boldsymbol{u}}[X_{\boldsymbol{v},\boldsymbol{\pi}(i)}]\Big] \geq \beta\Big\} \quad \vdash_d \quad -1 \geq 0.$$

The above definition is slightly more demanding than the most natural one, in which the hypotheses $X_{u,i}^2 = X_{u,i}$ are granted. As mentioned, in Section 9.3, we have established something noticeably stronger anyway: the following theorem is a restatement of Theorem 9.3.1:

Theorem 11.3.2. *Let $\mathcal{G} = \mathcal{G}(N, \eta) = (\mathcal{V}, \mathcal{E})$ be the Khot–Vishnoi instance of* UNIQUEGAMES *parameterized by N (a power of 2) and $\eta \in (0, 1)$, which has $2^N/N$ vertices, label-size N, and optimum value at most $N^{-\eta}$. Then there is a degree-4 SOS refutation that its fractional assignment optimum is at least $N^{-\Omega(\eta)}$.*

We now recall the KKMO [141] reduction from UG to MAXCUT, which is parameterized by $\rho \in (-1, 0)$. Given a UG instance $\mathcal{G}$ with label-size N, the reduction creates a vertex set V with a vertex $w_{u,x}$ for each $u \in \mathcal{V}$ and each $x \in \{-1, 1\}^N$. The probability distribution E on edges for the MAXCUT instance is given as follows:

- draw $\boldsymbol{u} \sim \mathcal{V}$;
- independently draw $(\boldsymbol{u}, \boldsymbol{v}_1, \boldsymbol{\pi}_1)$ and $(\boldsymbol{u}, \boldsymbol{v}_2, \boldsymbol{\pi}_2)$ from the marginal of $\mathcal{E}$ which has first vertex $\boldsymbol{u}$;
- draw "ρ-correlated strings" $(\boldsymbol{x}, \boldsymbol{y})$ from $\{-1, 1\}^N$;
- output the edge $(w_{\boldsymbol{u}_1, \boldsymbol{x} \circ \boldsymbol{\pi}_1}, w_{\boldsymbol{u}_2, \boldsymbol{y} \circ \boldsymbol{\pi}_2})$.

KKMO make the following easy observation:

Proposition 11.3.3. *Consider any cut $V \to \{-1, 1\}$ in the above-described* MAXCUT *instance (V, E); specifically, let us write it as a collection of functions $f_v : \{-1, 1\}^N \to \{-1, 1\}$, one for each $v \in \mathcal{V}$. Then the value of this cut is*

$$\tfrac{1}{2} - \tfrac{1}{2} \mathop{\mathbf{E}}_{\boldsymbol{u} \sim \mathcal{V}} [\mathbf{Stab}_\rho[g_u]],$$

where $g_u : \{-1, 1\}^N \to [-1, 1]$ is defined by $g_u(x) = \mathop{\mathbf{E}}_{(u, \boldsymbol{v}, \boldsymbol{\pi}) \sim \mathcal{E}|_{\boldsymbol{u}=u}} [f_{\boldsymbol{v}}(x \circ \boldsymbol{\pi})]$.

As mentioned, the KV MAXCUT instances are formed by composing the KKMO reduction with the KV UG instances. Khot and Vishnoi show that for any fixed $\eta \in (0, 1)$, the optimum value of the resulting MAXCUT instance is at most $(\arccos \rho)/\pi + o_N(1)$. Further, using "Majority cuts" it's easy to show (using, e.g. [179, Theorem 3.4.2]) that the optimum values is at *least* $(\arccos \rho)/\pi - o_N(1)$.

The main result of this section is the following:

Theorem 11.3.4. *Fix any small $\epsilon, \delta > 0$. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a UG instance with label-size N for which there is a degree-d SOS proof ($d \geq 2$) that its fractional assignment optimum is at most ϵ. Let $G = (V, E)$ be the* MAXCUT *instance resulting from applying the KKMO reduction with parameter $\rho \in (-1, 0)$ to $\mathcal{G}$. Then there is a degree $d + \tilde{O}(1/\delta^2)$ SOS refutation of the claim that the optimum value of G is at least $\frac{1}{2} - \frac{1}{\pi}\rho - (\frac{1}{2} - \frac{1}{\pi})\rho^3 + \delta + \epsilon \cdot 2^{\tilde{O}(1/\delta^2)}$.*

Together with Theorem 11.3.2 this implies:

Corollary 11.3.5. *Fix any small $\delta > 0$. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a Khot-Vishnoi UG instance with label-size N and noise parameter η. Let $G = (V, E)$ be the* MAXCUT *instance resulting from applying the KKMO reduction with parameter $\rho \in (-1, 0)$ to $\mathcal{G}$. Then there is a degree $\tilde{O}(1/\delta^2)$ SOS refutation of the claim that the optimum value of G is at least $\frac{1}{2} - \frac{1}{\pi}\rho - (\frac{1}{2} - \frac{1}{\pi})\rho^3 + \delta + 2^{\tilde{O}(1/\delta^2)} \cdot N^{-\Omega(\eta)}$.*

Corollary 11.3.6. *Consider the KV* MAXCUT *instances with parameter $\rho_0 \approx -.689$. The degree-$O(1)$ SOS SDP certifies they have value at most $.779$, which is within a factor $.952$ of the optimum. For general ρ, the degree-$O(1)$ SOS SDP certifies a value for the KV* MAXCUT *instances which is within a factor $.931$ of the optimum, where*

$$.931 \approx \min_{\rho \in (-1,0)} \frac{(\arccos \rho)/\pi}{\frac{1}{2} - \frac{1}{\pi}\rho - (\frac{1}{2} - \frac{1}{\pi})\rho^3}.$$

Before proving Theorem 11.3.4 we prove a lemma which gives an alternative way to refute a UG instance having a good solution: roughly, for most vertices $v \in \mathcal{V}$, its neighbors cannot agree well on what v's label should be.

Lemma 11.3.7. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a UG instance with label-size N and suppose there is a degree-d SOS refutation that its fractional assignment optimum is at least ϵ. Then also*

$$A \cup \left\{ \mathop{\mathbf{E}}_{\boldsymbol{u} \sim \mathcal{V}} \left[\sum_{i=1}^{N} \Big(\mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi}) \sim \boldsymbol{v}} [X_{\boldsymbol{v},\boldsymbol{\pi}(i)}] \Big)^2 \right] \geq 4\epsilon \right\} \quad \vdash_d \quad -1 \geq 0,$$

where

$$A = \{X_{u,i} \geq 0 : \forall x \in \mathcal{V}, i \in [N]\} \cup \{\sum_{i=1}^{N} X_{u,i} \leq 1 : \forall x \in \mathcal{V}\}.$$

Proof. Given the indeterminates $X_{u,i}$, define for each $u \in \mathcal{V}$ and $i \in [N]$ the homogeneous linear forms

$$Y_{u,i} = \tfrac{1}{2} X_{u,i} + \tfrac{1}{2} \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi}) \sim u} [X_{\boldsymbol{v},\boldsymbol{\pi}(i)}].$$

We will apply the assumption regarding the degree-d SOS refutation for $\mathcal{G}$ to the $Y_{u,i}$'s. Certainly we have

$$A \quad \vdash_1 \quad Y_{u,i} \geq 0, \sum_{j=1}^{N} Y_{u,j} \leq 1$$

for every $u \in \mathcal{V}$, $i \in [N]$. Indeed, it's not hard to check that to complete the proof we need only verify

$$A \quad \vdash_2 \quad \mathop{\mathbf{E}}_{\boldsymbol{u} \sim \mathcal{V}} \left[\sum_{i=1}^{N} Y_{\boldsymbol{u},i} \cdot \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi}) \sim \boldsymbol{u}} [Y_{\boldsymbol{v},\boldsymbol{\pi}(i)}] \right] \geq \tfrac{1}{4} \mathop{\mathbf{E}}_{\boldsymbol{u} \sim \mathcal{V}} \left[\sum_{i=1}^{N} \Big(\mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi}) \sim \boldsymbol{v}} [X_{\boldsymbol{v},\boldsymbol{\pi}(i)}] \Big)^2 \right].$$

But this follows from

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\boldsymbol{u}} \left[\sum_i Y_{\boldsymbol{u},i} \cdot \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi}) \sim \boldsymbol{u}} [Y_{\boldsymbol{v},\boldsymbol{\pi}(i)}] \right] \\
&= \tfrac{1}{2} \mathop{\mathbf{E}}_{\boldsymbol{u}} \left[\sum_i X_{\boldsymbol{u},i} \cdot \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi}) \sim \boldsymbol{u}} [Y_{\boldsymbol{v},\boldsymbol{\pi}(i)}] \right] + \tfrac{1}{2} \mathop{\mathbf{E}}_{\boldsymbol{u}} \left[\sum_i \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi}) \sim \boldsymbol{u}} [X_{\boldsymbol{v},\boldsymbol{\pi}(i)}] \cdot \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi}) \sim \boldsymbol{u}} [Y_{\boldsymbol{v},\boldsymbol{\pi}(i)}] \right] \\
&= \tfrac{1}{2} \mathop{\mathbf{E}}_{\boldsymbol{u}} \left[\sum_i X_{\boldsymbol{u},i} \cdot \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi}) \sim \boldsymbol{u}} [Y_{\boldsymbol{v},\boldsymbol{\pi}(i)}] \right] \\
&\qquad + \tfrac{1}{4} \mathop{\mathbf{E}}_{\boldsymbol{u}} \left[\sum_i \mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi}) \sim \boldsymbol{u}} [X_{\boldsymbol{v},\boldsymbol{\pi}(i)}] \cdot \mathop{\mathbf{E}}_{(\boldsymbol{v}',\boldsymbol{\pi}') \sim \boldsymbol{v}} [X_{\boldsymbol{v}',\boldsymbol{\pi}'(i)}] \right] + \tfrac{1}{4} \mathop{\mathbf{E}}_{\boldsymbol{u}} \left[\sum_i \Big(\mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi}) \sim \boldsymbol{u}} [X_{\boldsymbol{v},\boldsymbol{\pi}(i)}]^2 \Big) \right]
\end{aligned}$$

(where we do not even need the assumptions $\sum_{i=1}^{N} X_{u,i} \leq 1$). □

We now give the proof of Theorem 11.3.4.

Proof. It is not hard to deduce the following result from Corollary 11.2.5:

Corollary 11.3.8. *In the setting of Corollary 11.2.5, for any $\rho \in (-1, 0)$ we have*

$$A \quad \vdash_{\widetilde{O}(1/\delta^2)} \quad \mathbf{Stab}_\rho[f] \geq \tfrac{2}{\pi} \cdot \rho + (1 - \tfrac{2}{\pi}) \cdot \rho^3 - \delta - 2^{\widetilde{O}(1/\delta^2)} \cdot \sum_{i=1}^{n} \widehat{f}(i)^4.$$

It is also easy to check using Fact 8.3.3 that

$$\{f_v(x)^2 = 1 : \forall v \in V, x \in \{-1,1\}^N\} \quad \vdash_4 \quad g_v(x) \geq -1, g_v(x) \leq 1$$

for all $v \in V$, $x \in \{-1,1\}^N$. Thus using the above corollary and Proposition 11.3.3 we obtain

$$\begin{aligned}
&\{f_v(x)^2 = 1 : \forall v \in V, x \in \{-1,1\}^N\} \\
&\quad \vdash_{\widetilde{O}(1/\delta^2)} \quad \tfrac{1}{2} - \tfrac{1}{2} \mathop{\mathbf{E}}_{\boldsymbol{u} \sim \mathcal{V}} [\mathbf{Stab}_\rho[g_{\boldsymbol{u}}]] \leq \tfrac{1}{2} - \tfrac{1}{\pi}\rho - (\tfrac{1}{2} - \tfrac{1}{\pi})\rho^3 + \delta + 2^{\widetilde{O}(1/\delta^2)} \cdot \mathop{\mathbf{E}}_{\boldsymbol{u} \sim \mathcal{V}} \Big[\sum_{i=1}^{N} \widehat{g_{\boldsymbol{u}}}(i)^4 \Big].
\end{aligned} \tag{11.9}$$

Now we bound the error term $2^{\widetilde{O}(1/\delta^2)} \cdot \mathbf{E}_{\boldsymbol{u}\sim\mathcal{V}}[\sum_{i=1}^N \widehat{g_{\boldsymbol{u}}}(i)^4]$ as follows. Using the polynomial identity $\widehat{g_u}(i) = \mathbf{E}_{(\boldsymbol{v},\boldsymbol{\pi})\sim u}[\widehat{f_{\boldsymbol{v}}}(\boldsymbol{\pi}(i))]$ together with Fact 8.3.14 and Fact 8.3.15, we have

$$\vdash_4 \quad \mathop{\mathbf{E}}_{\boldsymbol{u}\sim\mathcal{V}}\Big[\sum_{i=1}^N \widehat{g_{\boldsymbol{v}}}(i)^4\Big] \leq \mathop{\mathbf{E}}_{\boldsymbol{u}\sim\mathcal{V}}\left[\sum_{i=1}^N \Big(\mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim\boldsymbol{u}}[\widehat{f_{\boldsymbol{v}}}(\boldsymbol{\pi}(i))^2]\Big)^2\right]. \tag{11.10}$$

On the other hand, it is easy to check that for all $i \in [N]$ and $v \in \mathcal{V}$, we have

$$\{f_v(x)^2 = 1 : \forall x \in \{-1,1\}^N\} \quad \vdash_2 \quad \widehat{f_v}(i)^2 \geq 0,\ \sum_{i=1}^N \widehat{f_v}(i)^2 \leq 1. \tag{11.11}$$

Since there is a degree-d refutation for $\mathcal{G}$ having a fractional assignment of value at least ϵ, implementing Lemma 11.3.7 with $X_{v,i} = \widehat{f_v}(i)^2$, we have

$$\begin{aligned}&\{\widehat{f_u}(i)^2 \geq 0 : \forall x \in \mathcal{V}, i \in [N]\} \cup \{\sum_{i=1}^N \widehat{f_u}(i)^2 \leq 1 : \forall x \in \mathcal{V}\}\\ &\qquad \cup \left\{\mathop{\mathbf{E}}_{\boldsymbol{u}\sim\mathcal{V}}\left[\sum_{i=1}^N \Big(\mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim\boldsymbol{u}} \widehat{f_{\boldsymbol{v}}}(\boldsymbol{\pi}(i))^2\Big)^2\right] \geq 4\epsilon\right\} \quad \vdash_{d+2} \quad -1 \geq 0.\end{aligned} \tag{11.12}$$

By Fact 8.2.3, (11.11) and (11.12) give

$$\begin{aligned}&\{f_v(x)^2 = 1 : \forall v \in \mathcal{V}, x \in \{-1,1\}^N\} \cup \left\{\mathop{\mathbf{E}}_{\boldsymbol{u}\sim\mathcal{V}}\left[\sum_{i=1}^N \Big(\mathop{\mathbf{E}}_{(\boldsymbol{v},\boldsymbol{\pi})\sim\boldsymbol{u}} \widehat{f_{\boldsymbol{v}}}(\boldsymbol{\pi}(i))^2\Big)^2\right] \geq 4\epsilon\right\}\\ &\qquad \vdash_{d+4} \quad -1 \geq 0.\end{aligned} \tag{11.13}$$

Combining (11.13) and (11.10), we get

$$\{f_v(x)^2 = 1 : \forall v \in \mathcal{V}, x \in \{-1,1\}^N\} \cup \left\{\mathop{\mathbf{E}}_{\boldsymbol{u}\sim\mathcal{V}}\Big[\sum_{i=1}^N \widehat{g_{\boldsymbol{v}}}(i)^4\Big] \geq 4\epsilon\right\} \quad \vdash_{d+2} \quad -1 \geq 0. \tag{11.14}$$

Finally, combining (11.14) and (11.9), we get

$$\begin{aligned}&\{f_v(x)^2 = 1 : \forall v \in \mathcal{V}, x \in \{-1,1\}^N\}\\ &\cup\{\tfrac{1}{2}-\tfrac{1}{2}\mathop{\mathbf{E}}_{\boldsymbol{u}\sim\mathcal{V}}[\mathbf{Stab}_\rho[g_{\boldsymbol{u}}]] \geq \tfrac{1}{2}-\tfrac{1}{\pi}\rho-(\tfrac{1}{2}-\tfrac{1}{\pi})\rho^3+\delta+2^{\widetilde{O}(1/\delta^2)}\} \quad \vdash_{d+\tilde{O}(1/\delta^2)} \quad -1 \geq 0. \quad \square\end{aligned}$$

11.4 An approximator for the absolute-value function

Here we restate and prove Lemma 11.2.1. A key tool will be the polynomial approximator to the sgn function constructed in [80].

Lemma 11.2.1. *For any sufficiently small parameter $\delta > 0$, there exists a univariate, real, even polynomial $P(t) = Q(t^2)$ of degree at most $\widetilde{O}(1/\delta^2)$ such that:*

1. *$P(t) \geq |t|$ for all $t \in \mathbb{R}$;*
2. *$\mathbf{E}[P(\sigma \boldsymbol{g})] \leq \sqrt{\frac{2}{\pi}} \cdot \sigma + \delta \leq \left(\frac{1}{2}\sigma^2 + \frac{1}{\pi}\right) + \delta$ for all $0 \leq \sigma \leq 1$, where $\boldsymbol{g} \sim \mathrm{N}(0,1)$;*
3. *Each coefficient of P is at most $2^{O(\deg(P))}$ in absolute value.*

Proof. We will use the following result from [80, Theorem 3.10]:

Theorem 11.4.1. *For every $0 < \epsilon < .1$ there is an odd integer $d = d(\epsilon) = \Theta(\log^2(1/\epsilon)/\epsilon)$ and a univariate polynomial $p(t)$ of degree d satisfying:*

- *$p(t) \in [\mathrm{sgn}(t) - \epsilon, \mathrm{sgn}(t) + \epsilon]$ for all $|t| \in [\epsilon, 1]$;*
- *$p(t) \in [-1 - \epsilon, 1 + \epsilon]$ for all $|t| \leq \epsilon$;*
- *$p(t)$ is monotonically increasing on the intervals $(-\infty, -1]$ and $[1, +\infty)$.*

We can assume without loss of generality that $p(t)$ is odd since the odd part of $p(t)$ (i.e. $(p(t) - p(-t))/2$) also satisfies the properties in Theorem 11.4.1.

Given $p(t)$ as in Theorem 11.4.1, define

$$p_0(t) = (1 + 2\epsilon)p(t/M), \quad \text{where } M = \frac{c \log^c(1/\epsilon)}{\sqrt{\epsilon}}$$

and $c > 1$ is a universal constant to be chosen later. The polynomial $p_0(t)$ has the following properties:

- $p_0(t) \in [1, 1{+}4\epsilon]$ when $t \in [M\epsilon, M]$, $p_0(t) \in [-(1{+}4\epsilon), -1]$ when $t \in [-M, -M\epsilon]$;
- $p_0(t) \in [-(1 + 4\epsilon), 1 + 4\epsilon]$ for all $|t| \leq M\epsilon$;
- $p_0(t) \geq 1$ when $t \geq M$, $p_0(t) \leq 1$ when $t \leq -M$.

Finally, define

$$P(t) = \int_0^t p_0(x)dx + 2M\epsilon.$$

an even polynomial of degree $d+1$. We will show that the following hold assuming c is taken sufficiently large and then ϵ is sufficiently small:

(a) $P(t) \geq |t|$ for all $t \in \mathbb{R}$;

(b) $\mathbf{E}[P(\sigma \boldsymbol{g})] \leq \sqrt{\frac{2}{\pi}} \cdot \sigma + O(M\epsilon)$ for all $0 \leq \sigma \leq 1$;

(c) Each coefficient of P is at most $2^{O(d)}$ in absolute value.

The proof is then completed by taking $\epsilon = \delta^2/\mathrm{polylog}(1/\delta)$.

Properties ((a)) follows easily from the definition of $P(t)$. It also follows easily from the definition that $|P(t)| \leq 1 + O(M\epsilon) \leq 2$ for all $|t| \leq 1$. It is a standard fact in approximation theory (see, e.g., [208, 176]) that if P is a degree $d+1$ polynomial satisfying $|P(t)| \leq b$ for all $|t| \leq 1$ then each coefficient of $P(t)$ is at most, say, $b(4e)^{d+1} = 2^{O(d)}$ in magnitude. This verifies ((c)). It remains to establish property ((b)). For this we have

$$\mathbf{E}[P(\sigma \boldsymbol{g})] = \mathbf{E}[P(\sigma \boldsymbol{g}) \cdot \mathbf{1}\{|\sigma \boldsymbol{g}| \leq M\}] + \mathbf{E}[P(\sigma \boldsymbol{g}) \cdot \mathbf{1}\{|\sigma \boldsymbol{g}| > M\}] \tag{11.15}$$

Regarding the first term in (11.15) we use that for $|t| \leq M$ we have $|p_0(t)| \leq 1 + 4\epsilon$ and hence

$$P(t) \leq (1+4\epsilon)|t| + 2M\epsilon = |t| + O(M\epsilon) \quad \forall |t| \leq M. \tag{11.16}$$

Thus

$$\begin{aligned} \mathbf{E}[P(\sigma \boldsymbol{g}) \cdot \mathbf{1}\{|\sigma \boldsymbol{g}| \leq M\}] &\leq \mathbf{E}[|\sigma \boldsymbol{g}| \cdot \mathbf{1}\{|\sigma \boldsymbol{g}| \leq M\}] + O(M\epsilon) \\ &\leq \mathbf{E}[|\sigma \boldsymbol{g}|] + O(M\epsilon) = \sqrt{\tfrac{2}{\pi}} \cdot \sigma + O(M\epsilon). \end{aligned}$$

To complete the verification of ((b)) it therefore suffices to bound the second term in (11.15) by $O(M\epsilon)$. In fact we will show

$$\mathbf{E}[P(\sigma \boldsymbol{g}) \cdot \mathbf{1}\{|\sigma \boldsymbol{g}| > M\}] \ll M\epsilon. \tag{11.17}$$

Using evenness of P and the fact that it is evidently increasing on $[M, \infty)$ we have

$$\mathbf{E}[P(\sigma \boldsymbol{g}) \cdot \mathbf{1}\{|\sigma \boldsymbol{g}| > M\}] = 2\,\mathbf{E}[P(\sigma \boldsymbol{g}) \cdot \mathbf{1}\{\sigma \boldsymbol{g} > M\}] \leq 2\,\mathbf{E}[P(\boldsymbol{g}) \cdot \mathbf{1}\{\boldsymbol{g} > M\}]. \tag{11.18}$$

We upper-bound P's value on large inputs using a well-known fact from approximation theory (and a corollary of the theorem in §33 of [169]):

Fact 11.4.2. *Let $q(t)$ be a polynomial of degree at most k satisfying $|q(t)| \leq b$ for all $|t| \leq 1$. Then $|q(t)| \leq b\,|3t|^k$ for all $|t| \geq 1$.*

Applying this fact to $p(t)$ we obtain $p(t) \leq (1+\epsilon)(3t)^d$ for all $t \geq 1$, whence $p_0(t) \leq 2(3t/M)^d$ for all $t \geq M$, whence

$$P(t) = \int_0^t p_0(x)dx + 2M\epsilon \leq O(M) + t \cdot 2(3t/M)^d \leq O(1) \cdot (3t/M)^{d+1}$$

for all $t \geq M$ (we also used $M = o(2^d)$). Thus

$$\begin{aligned}(11.18) &\leq O(\tfrac{3}{M})^{d+1} \cdot \mathbf{E}[\boldsymbol{g}^{d+1} \cdot \mathbf{1}\{\boldsymbol{g} > M\}] \leq O(\tfrac{3}{M})^{d+1} \cdot O(dM)^{d+1} \exp(-M^2/2) \\ &\leq 2^{\mathrm{polylog}(1/\epsilon)/\epsilon} \exp(-M^2/2) = 2^{\mathrm{polylog}(1/\epsilon)/\epsilon} \exp(-c^2 \log^{2c}(1/\epsilon)/2\epsilon) \ll M\epsilon\end{aligned}$$

if we choose c to be a large enough universal constant. This completes the justification of (11.17) and the overall proof. $\square$

Part III

Robust algorithms

Chapter 12

Robust satisfiability algorithms for width 1 CSPs

12.1 Introduction

Constraint satisfaction problems (CSPs) constitute a broad and important subclass of algorithmic tasks. One approach to studying the complexity of CSPs centers around the Feder–Vardi *Dichotomy Conjecture* [86] and the use of algebra [125] to classify all CSP decision problems. Another approach to the study of CSPs involves quantifying the extent to which natural CSPs can be *approximately* solved [127]; this approach has been characterized by more "analytic" methods. Recently there has been interest in melding the two approaches (see, e.g., [155, 128, 113]); the present work takes another step in this direction.

Almost-satisfiable instances. The algebraic approach to CSPs is mainly concerned with what we'll call the *decision* problem for CSPs: given an instance, is it completely satisfiable? The Dichotomy Conjecture states that for every CSP this task is either in P or is NP-hard; the *Algebraic Dichotomy Conjecture* of Bulatov, Jeavons, and Krokhin [53] refines this by conjecturing a precise algebraic characterization of the tractable CSP decision problems. However when it comes to approximability, not all tractable CSPs are "equally tractable". E.g., for Max-Cut, not only can one efficiently find a completely satisfying assignment when one exists, the Goemans–Williamson algorithm [94] efficiently finds an *almost*-satisfying assignment whenever an *almost*-satisfying assignment exists. (Specifically, it finds a $(1 - O(\sqrt{\epsilon}))$-satisfying assignment whenever a $(1 - \epsilon)$-satisfying assignment exists.) Contrast this with the kLin(mod 2) problem, $k \geq 3$: again, one can efficiently find a completely satisfying assignment whenever one exists; however Håstad [116] has shown that finding even a *somewhat*-satisfying assignment whenever an *almost*-satisfying assignment exists is NP-hard. (Specifically, $\forall \epsilon > 0$ it is hard to find a $(1/2 + \epsilon)$-satisfying assignment when a $(1 - \epsilon)$-satisfying assignment exists.)

Prior work on robust decidability. In 1997, Zwick [228] initiated the study of the following very natural problem: which CSPs are efficiently *robustly decidable*? By this we mean that the algorithm should find $(1 - o_\epsilon(1))$-satisfying assignments whenever $(1 - \epsilon)$-satisfying assignments exist (formal definitions are given in Section 12.2). Zwick gave a linear programming (LP)-based algorithm for finding $(1 - O(1/\log(1/\epsilon)))$-satisfying assignments for Horn-kSat (for any fixed k); he also gave a semidefinite programming (SDP)-based algorithm for finding $(1 - O(\epsilon^{1/3}))$-satisfying assignments for 2Sat (since improved to $1 - O(\epsilon^{1/2})$ [60]). Later, Khot [136] gave an SDP-based algorithm for finding $(1 - \tilde{O}(\epsilon^{1/5}))$-satisfying assignments for the notorious *Unique-Games* problem over domains D with $|D| = O(1)$ (since improved to $1 - O(\epsilon^{1/2})$ [59]).[1] On the other hand, the only tractable CSPs for which the robust decision problem seems to be NP-hard are the ones that can encode linear equations over groups.

Bounded width. If we wish to classify the CSPs which are efficiently robustly decidable, we seek a property that is shared by Horn-kSat, 2Sat, and Unique-Games but not by 3Lin(mod p). From the algebraic viewpoint on CSPs there is a very obvious candidate: the former CSPs have *bounded width* while the latter does not. Briefly, a CSP is said to have *bounded width* if unsatisfiable instances can always be "refuted" in a proof system that only allows for *constant*-sized partial assignments to be kept "in memory" (again, more formal definitions are in Section 12.2). Recent independent works of Barto–Kozik [37] and Bulatov [52] have connected this notion to algebra by showing that bounded-width CSPs coincide with those which cannot encode linear equations over groups. Thus by Håstad's work we know that any CSP which is efficiently robustly decidable must have bounded width (assuming $\mathsf{P} \neq \mathsf{NP}$). As mentioned at the beginning of this thesis, the Guruswami–Zhou Conjecture [113] states that the converse also holds: every bounded-width CSP has an efficient robust decision algorithm.

[1]We emphasize that in this chapter, we always treat the domain size $|D|$ as a fixed constant, with $\epsilon \to 0$ independently.

Linear and semidefinite programming. Essentially the only known way to produce CSP approximation algorithms is through the use of LPs and SDPs. Indeed, recent work of Raghavendra [189] shows that if one believes Khot's Unique Games Conjecture [136], then a CSP Π is efficiently robustly decidable if and *only if* the basic SDP relaxation robustly decides it. However understanding and solving SDPs can be difficult, and as Zwick's Horn-kSat algorithm illustrates, sometimes only the power of linear programming is needed for robust decision algorithms.

12.1.1 Our contributions

As a step towards the Guruswami–Zhou Conjecture, we show that a special case of the bounded width CSPs are robustly decidable by the basic *linear programming relaxation*. Somewhat informally stated, our main theorem is the following:

Theorem 12.1.1 (Informal version of Theorem 12.3.1)**.** *Let* Π *be any (finitely presented) CSP. Then the basic LP relaxation robustly decides* Π *if* Π *has* width 1.

(Formal definitions of the terms in this theorem are given in Section 12.2.)

In slightly more details, our proof of Theorem 12.1.1 gives an efficient deterministic "LP-rounding" algorithm for actually *finding* the required almost-satisfying assignments. Quantitatively, it finds $(1 - O(1/\log(1/\epsilon)))$-satisfying assignments for $(1 - \epsilon)$-satisfiable instances, matching the performance of Zwick's Horn-kSat algorithm. As we describe below, this is best possible. Our rounding algorithm is also simpler than Zwick's.

Independently and concurrently, Dalmau and Krokhin have also shown that width 1 CSPs have efficient robust decision algorithms. Their proof is different from ours; it is by a black-box reduction to Zwick's Horn-Sat algorithm.

The quantitative dependence on ϵ. As mentioned, our LP-based algorithm for width-1 CSPs finds $(1 - O(1/\log(1/\epsilon)))$-satisfying assignments to $(1 - \epsilon)$-satisfiable instances. One might hope for a better (say, polynomial) dependence on ϵ here. Unfortunately, this is not possible. Zwick [228] already showed that for Horn-3Sat there are "gap instances" where the basic LP has value $1-\epsilon$ but the optimum value is only $1-\Omega(1/\log(1/\epsilon))$. Indeed Guruswami and Zhou [113] extended this by showing there are equally bad gap instances for the basic SDP relaxation of Horn-3Sat. Assuming the Unique Games Conjecture, Raghavendra's work [189] in turn implies that *no* polynomial-time algorithm can find $(1 - o(1/\log(1/\epsilon)))$-satisfying assignments to $(1 - \epsilon)$-satisfiable instances. On a positive note, in Section 12.3.1 we show that for the special case of width-1 CSPs called "lattice CSPs", the basic LP relaxation can be used to find $(1 - O(\epsilon))$-satisfying assignments to $(1 - \epsilon)$-satisfiable instances.

12.2 Preliminaries

12.2.1 CSP preliminaries

Definitions. Let D be a nonempty finite *domain* of values, and let Γ be a nonempty finite set of *relations* over D, each of positive finite arity. We write such a k-ary relation as $R : D^k \to \{0, 1\}$. An *instance* $\mathcal{I}$ of the *constraint satisfaction problem* $CSP(\Gamma)$ consists of a set V of n *variables*, along with a list of m *constraints*. Each constraint C is a pair (S, R), where S is a tuple of some k variables (the *scope* of the constraint), and R is a k-ary relation in the set Γ. We say that $\mathcal{I}'$ is a *sub-instance* of $\mathcal{I}$ if contains just a subset of the variables and constraints in $\mathcal{I}$; it is *induced* by the variable set $V' \subseteq V$ if it includes all constraints in $\mathcal{I}$ involving just the variables in V'. An *assignment* for an instance of $CSP(\Gamma)$ is any mapping $\alpha : V \to D$. The assignment *satisfies* a constraint $C = (S, R)$ if $R(\alpha(S)) = 1$ (where α operates on S_i component-wise). The *value* of the assignment, $\mathsf{val}_{\mathcal{I}}(\alpha) \in [0, 1]$, is the fraction of constraints it satisfies. We define the *optimum value* of the instance $\mathcal{I}$ to be $\mathsf{opt}(\mathcal{I}) = \max_\alpha\{\mathsf{val}_{\mathcal{I}}(\alpha)\}$. We say the instance is *satisfiable* if $\mathsf{opt}(\mathcal{I}) = 1$.

CSP width. An important parameter of a $CSP(\Gamma)$ problem is its *width*. This notion, dating back to Feder and Vardi [86], can be given many equivalent definitions (in terms of, e.g., pebble games, Datalog, logic, tree-width, proof complexity...). Roughly speaking, $CSP(\Gamma)$ has width k if unsatisfiable instances of $CSP(\Gamma)$ can always be refuted while only keeping k partial assignments "in memory". More formally, given an instance $\mathcal{I}$ of $CSP(\Gamma)$, consider the following (k, ℓ) *pebble game* with $1 \leq k < \ell$ integers: Alice begins by placing each of ℓ pebbles on variables in V. Bob must respond with a partial assignment to the pebbled variables which satisfies all constraints in which they participate. On each subsequent turn, Alice may move $\ell - k$ of the pebbles to different vertices. Bob must respond with a partial assignment to the newly pebbled variables which again satisfies all constraints in which the pebbled variables participate, and which is consistent with the assignment to the k unmoved pebbles from the previous turn. If ever Bob cannot respond, Alice wins the game; if Bob can always play forever, he wins the game. If $\mathcal{I}$ is a satisfiable instance then Bob can always win regardless of k and ℓ; on the other hand, if $\mathcal{I}$

is unsatisfiable, then Alice may or may not be able to win. We say that $CSP(\Gamma)$ has *width* (k, ℓ) if Alice can win the (k, ℓ) pebble game on all unsatisfiable instances; and, we say that $CSP(\Gamma)$ has *width* k if it has width (k, ℓ) for some finite ℓ. In particular, we say that $CSP(\Gamma)$ has *bounded width* if it has width k for some finite k. Bounded width CSPs can be solved in polynomial time using a simple enumeration over Bob's possible strategies. As examples, Horn-kSat has width 1, 2-Colorability has width 2 (but not width 1), and 3Lin(mod 2) does not have bounded width.

Tree duality and width 1. It is well known [86] that the CSPs of width 1 can be precisely characterized as those which have *tree duality*. We say that $CSP(\Gamma)$ has tree duality if for every unsatisfiable instance $\mathcal{I}$ there is a unsatisfiable "tree" instance $\mathcal{T}$ which "witnesses" this. By "witness" we mean that there is a *homomorphism* from $\mathcal{T}$ to $\mathcal{I}$; i.e., a map from $\mathcal{T}$'s variables into $\mathcal{I}$'s variables which preserves all relations. The definition of a "tree" instance is the natural one in case all relations in Γ have arity 2; in general, we must make more careful definitions. We define a *walk* in instance $\mathcal{I}$ of $CSP(\Gamma)$ to be a sequence $x_1, C_1 = (S_1, R_1), t_1, u_1, x_2, C_2 = (S_2, R_2), t_2, u_2, \ldots, x_{\ell+1}$ where each x_i is a variable in $\mathcal{I}$, each C_i is a constraint in $\mathcal{I}$, the indices t_i and u_i are distinct, and $(S_i)_{t_i} = x_i$, $(S_i)_{u_i} = x_{i+1}$ for all $i \in [\ell]$. We say the walk proceeds from *starting point* x_1 to *endpoint* x_ℓ. We say the walk is *non-backtracking* if for every $i \in [\ell]$ either C_i differs from C_{i+1} or $u_i \neq t_{i+1}$. We say that $\mathcal{I}$ is *connected* if there is a walk from x to y for all pairs of distinct variables x and y in $\mathcal{I}$. Finally, we say that $\mathcal{I}$ is a *tree* if it is connected and it does not contain any non-backtracking walk with the same starting point and endpoint.

12.2.2 Algorithmic preliminaries

Approximation algorithms. For real numbers $0 \leq s \leq c \leq 1$, we say an algorithm (c, s)-*approximates* $CSP(\Gamma)$ if it outputs an assignment with value at least s on any input instance with value at least c. For $c = s = 1$ we simply say that the algorithm *decides* $CSP(\Gamma)$; this means the algorithm always finds a satisfying assignment given a satisfiable instance. We say that an algorithm *robustly decides* $CSP(\Gamma)$ if there is an *error function* $r : [0, 1] \to [0, 1]$ with $r(\epsilon) \to 0$ as $\epsilon \to 0$ such that the algorithm $(1 - \epsilon, 1 - r(\epsilon))$-approximates $CSP(\Gamma)$ for all $\epsilon \in [0, 1]$. In particular, the algorithm must decide $CSP(\Gamma)$.

The basic integer program. For any instance $\mathcal{I}$ of $CSP(\Gamma)$ there is an equivalent canonical 0-1 integer program we denote by $\mathrm{IP}(\mathcal{I})$. It has variables $p_v(j)$ for each $v \in V$, $j \in D$, as well as variables $q_{C_i}(J)$ for each arity-k constraint $C_i = (S_i, R_i)$ and tuple

$J \in D^{k_i}$. The interpretation of $p_v(j) = 1$ is that variable v is assigned value j; the interpretation of $q_{C_i}(J) = 1$ is that the k_i-tuple of variables S_i is assigned the k_i-tuple of values J. More formally, $\mathrm{IP}(\mathcal{I})$ is the following:

$$\text{maximize} \quad \frac{1}{m}\sum_{i=1}^{m}\sum_{J:R_i(J)=1} q_{C_i}(J)$$

$$\text{subject to:} \quad \sum_{j \in D} p_v(j) = 1 \qquad \text{for all } v \in V, \tag{12.1}$$

$$\sum_{J \in D^{k_i}:J_t=j} q_{C_i}(J) = p_v(j) \qquad \text{for all } C_i \text{ and } t \text{ such that } (S_i)_t = v. \tag{12.2}$$

The optimum value of $\mathrm{IP}(\mathcal{I})$ is precisely $\mathsf{opt}(\mathcal{I})$. Note that the size of this integer programming formulation is $\mathrm{poly}(n, m)$ (as we are assuming that D and Γ are of constant size).

The basic linear program. If we relax $\mathrm{IP}(\mathcal{I})$ by having the variables take values in the range $[0, 1]$ rather than $\{0, 1\}$, we obtain the *basic linear programming relaxation* which we denote by $\mathrm{LP}(\mathcal{I})$. An optimal solution of $\mathrm{LP}(\mathcal{I})$ can be computed in $\mathrm{poly}(n, m)$ time; the optimal value, which we denote by $\mathsf{opt}^{\mathrm{LP}}(\mathcal{I})$, always satisfies $\mathsf{opt}(\mathcal{I}) \leq \mathsf{opt}^{\mathrm{LP}}(\mathcal{I}) \leq 1$. We interpret any feasible solution to $\mathrm{LP}(\mathcal{I})$ as follows: For each $v \in V$, the quantities $p_v(j)$ form a discrete probability distribution on D (because of (12.1)), denoted p_v. For each k_i-ary constraint $C_i = (S_i, R_i)$, the quantities $q_{C_i}(J)$ form a probability distribution on D^{k_i}, denoted q_{C_i}. Furthermore (because of (12.2)), the marginals of the q_{C_i} distributions are "consistent" with the p_v distributions, in the sense that whenever $(S_i)_t = v$ it holds that $\Pr_{J \sim q_{C_i}}[J_t = j] = p_v(j)$ for all $j \in D$. Finally, the objective value to be optimized in $\mathrm{LP}(\mathcal{I})$ is

$$\mathsf{val}^{\mathrm{LP}}_{\mathcal{I}}(\{p_v\}, \{q_{C_i}\}) = \frac{1}{m}\sum_{i=1}^{m} \Pr_{J \sim q_{C_i}}[R_i(J) = 1];$$

the optimum value of this over all feasible solutions is $\mathsf{opt}^{\mathrm{LP}}(\mathcal{I})$.

12.2.3 Algebraic preliminaries

Polymorphisms. The Dichotomy Conjecture of Feder and Vardi [86] asserts that for each Γ, the problem of deciding $CSP(\Gamma)$ is either in P or is NP-complete. The most successful approach towards this conjecture has been the algebraic one initiated by Jeavons and coauthors [125] in which the problem is studied through the *polymorphisms* of Γ. We say $f : D^\ell \to D$ is an ℓ-ary polymorphism for the k-ary relation R if $R(f(x^1), \dots, f(x^k)) = 1$ whenever $R(x_i^1, \dots, x_i^k) = 1$ for all $i \in [\ell]$ (here each x^j is a tuple in D^ℓ). We say that f is a polymorphism for Γ if it is a polymorphism for each relation in Γ. We say that Γ is a *core*, if all of its 1-ary polymorphisms are bijections (at a high level, this means that there are no superfluous values in D for $CSP(\Gamma)$). Finally, we call a polymorphism f *idempotent*, if $f(j, \dots, j) = j$ for all $j \in D$.

Polymorphisms and width. Recently, independent works of Barto–Kozik [37] and Bulatov [52] managed to characterize bounded-width CSPs in terms of their polymorphisms. Specifically, they showed that $CSP(\Gamma)$ has bounded width (for Γ a core) if and only if Γ has an ℓ-ary *weak near-unanimity (WNU)* polymorphism for all $\ell \geq 3$. Here a polymorphism f is said to be WNU if it is idempotent and has the following symmetry: $f(x, x, \dots, x, y) = f(x, \dots, x, y, x) = \cdots = f(x, y, x, \dots, x) = f(y, x, x, \dots, x)$.

Much earlier, Dalmau and Pearson [73] gave a straightforward characterization the class of width-1 CSPs in terms of their polymorphisms. Specifically, they showed that $CSP(\Gamma)$ has width 1 if and only if Γ is preserved by a *set operation* $g : \mathcal{P}(D) \to D$. This means that $f : D^\ell \to D$ defined by $f(x_1, \dots, x_\ell) = g(\{x_1, \dots, x_\ell\})$ is a polymorphism for all $\ell \geq 1$. Note that all these polymorphisms are *symmetric*, meaning invariant under all permutations of the inputs. We will use a simple lemma about width-1 CSPs which first requires a definition.

Definition 12.2.1. *Let $\mathcal{J}$ be a subset of a cartesian product $B_1 \times \cdots \times B_k$ of nonempty sets. We say $\mathcal{J}$ is* subdirect, *written $\mathcal{J} \subseteq_S B_1 \times \cdots \times B_k$, if for each $i \in [k]$ the projection of $\mathcal{J}$ to the i'th coordinate is all of B_i.*

Lemma 12.2.2. *Say g is a set operation for $CSP(\Gamma)$, R is an arity-k relation in Γ, and $B_1, \dots, B_k \subseteq D$. Assume there is a $\mathcal{J} \subseteq_S B_1 \times \cdots \times B_k$ all of whose members satisfy R. Then $R(g(B_1), \dots, g(B_k)) = 1$.*

Proof. For each $t \in [k]$ and $j \in B_t$, select some $J^{t,j} \in \mathcal{J}$ whose t'th coordinate is j.. Think of the $\ell = \sum |B_t| \geq 1$ tuples $J^{t,j}$ as column vectors, and adjoin them in some order to form a $k \times \ell$ matrix X. Let x^t be the t^{th} row of X. It is clear that the set of values appearing in x^t is precisely B_t. Thus if f is the ℓ-ary polymorphism defined by g, we have $f(x^t) = g(B_t)$. But since f is a polymorphism and each $J^{t,j}$ satisfies R, it follows that $R(g(B_1), \dots, g(B_k)) = 1$. $\square$

12.3 Width 1 implies robust decidability by LP

The following theorem shows that a simple rounding algorithm for the basic linear program robustly decides any width-1 CSP.

Theorem 12.3.1. *Let Γ be a finite set of relations over the finite domain D, each relation having arity at most K. Assume that $CSP(\Gamma)$ has width 1. Then there is a $\mathrm{poly}(n,m)$-time algorithm for $CSP(\Gamma)$ which when given an input $\mathcal{I}$ with $\mathsf{val}^{\mathrm{LP}}(\mathcal{I}) = 1-\epsilon$ outputs an assignment $\alpha : V \to D$ with $\mathsf{val}_{\mathcal{I}}(\alpha) \geq 1 - O(K^2|D|\log(2|D|))/\log(1/\epsilon)$. (In particular, $\mathsf{val}_{\mathcal{I}}(\alpha) = 1$ if $\mathsf{val}^{\mathrm{LP}}(\mathcal{I}) = 1$.)*

Proof. The first step of the algorithm is to solve the LP relaxation of the instance, determining an optimal solution $\{p_v : v \in V\}$, $\{q_{C_i} : i \in [m]\}$ which obtains $\mathsf{val}^{\mathrm{LP}}(\mathcal{I}) = 1 - \epsilon$. For technical reasons we will now assume without loss of generality that $K \geq 2$ and that

$$2^{-\mathrm{poly}(n,m)} \leq \epsilon \leq \frac{1}{4(2|D|)^{2(K-1)}}. \tag{12.3}$$

The assumption (12.3) is also without loss of generality. We may assume the upper bound by adjusting the constant in the $O(\cdot)$ of our theorem. As for the lower bound, since linear programming is in polynomial time, ϵ will be either 0 or at least $2^{-\mathrm{poly}(n,m)}$. In the former case, we replace ϵ with a sufficiently small $2^{-O(m)}$ so that the theorem's claimed lower bound on $\mathsf{val}_{\mathcal{I}}(\alpha)$ exceeds $1 - 1/m$; then $\mathsf{val}_{\mathcal{I}}(\alpha) > 1 - 1/m$ implies $\mathsf{val}_{\mathcal{I}}(\alpha) = 1$ as required when $\mathsf{val}^{\mathrm{LP}}(\mathcal{I}) = 1$.

For a particular constraint C_i, let $\epsilon_i = \sum_{J:R_i(J)=0} q_{C_i}(J)$. Since $\mathsf{val}^{\mathrm{LP}}(\mathcal{I}) \geq 1 - \epsilon$ we have $\mathrm{avg}\{\epsilon_i\} \leq \epsilon$. The next step is to "give up" on any constraint having $\epsilon_i > \sqrt{\epsilon}$. By Markov's inequality the fraction of such constraints is at most $\sqrt{\epsilon}$, which is negligible compared to the $O(1/\log(1/\epsilon))$ error guarantee of our algorithm. For notational simplicity, we now assume that $\epsilon_i \leq \sqrt{\epsilon}$ for all $i \in [m]$.

We now come to the main part of the algorithm. Since $CSP(\Gamma)$ has width 1, it has a set operation $g : \mathcal{P}(D) \to D$. We first describe a simple *randomized* "LP-rounding" algorithm based on g:

1. Let $r = (2|D|)^{K-1}$ and let $b = \lfloor \log_r(1/2\sqrt{\epsilon}) \rfloor$. We have $b \geq 1$ by (12.3).
2. Choose $\theta \in \{r^{-1}, r^{-2}, \ldots, r^{-b}\}$ uniformly at random. Note that $r^{-b} \geq 2\sqrt{\epsilon}$.
3. Output the assignment $\alpha : V \to D$ defined by $\alpha(v) = g(\mathrm{supp}_\theta(p_v))$, where $\mathrm{supp}_\theta(p_v)$ denotes $\{j \in D : p_v(j) \geq \theta\}$.

We will show for each constraint $C_i = (S_i, R_i)$ that

$$\Pr[R_i(\alpha(S_i)) = 0] \leq K|D|/b. \tag{12.4}$$

It follows from linearity of expectation that the expected fraction of constraints not satisfied by α is at most $K|D|/b = O(K^2|D|\log(2|D|))/\log(1/\epsilon)$. This would complete the proof, except for the fact that we have given a randomized algorithm. However we can easily make the algorithm deterministic and efficient by trying all choices for θ (of which there are at most $b \leq \mathrm{poly}(n, m)$ by (12.3)) and selecting the best resulting assignment.

We now give the analysis justifying (12.4) for each fixed constraint $C_i = (S_i, R_i)$. For simplicity we henceforth write $C = C_i$, $S = S_i$, $R = R_i$ and suppose that R has arity $k \leq K$. Let us say that a choice of θ is *bad* if it falls into the interval $(p_{S_t}(j)/r, p_{S_t}(j)]$ for some $t \in [k]$ and $j \in D$. For each choice of t and j there is at most one bad choice of θ for the associated interval; hence the overall probability θ is bad is at most $K|D|/b$. Thus it suffices to show that whenever θ is not bad, C is satisfied by α.

For each $t \in [k]$ let $B_t = \mathrm{supp}_\theta(p_{S_t})$; these sets are nonempty because $\theta \leq r^{-1} \leq |D|^{-1}$. Also, let $\mathcal{J} = \{J \in B_1 \times \cdots \times B_k : R(j) = 1\}$. By Lemma 12.2.2, to show that C is satisfied by α, we only need to show that $J \subseteq_S B_1 \times \cdots \times B_k$ — i.e., that for all $t \in [k]$ and all $j \in B_t$ there exists a tuple $J \in \mathcal{J}$ such that $J_t = j$. We show this is true for $t = k$ and the statement for other values of t follows in the same way. For any $j \in B_k$, we have $\theta \leq p_{S_k}(j)$ by the definition of B_t. Since θ is not bad, we know that $\theta \notin (p_{S_k}(j)/r, p_{S_k}(j)]$. Therefore we have $\theta \leq p_{S_k}(j)/r$. Now since all but at most $\epsilon_i \leq \sqrt{\epsilon}$ of the probability mass in q_C is on assignments satisfying R, we conclude

$$\sum_{J' \in D^{k-1} : R(J', j) = 1} q_C(J', j) \geq p_{S_k}(j) - \sqrt{\epsilon} \geq p_{S_k}(j)/2.$$

Here we used $2\sqrt{\epsilon} \leq r^{-b} \leq \theta \leq p_{S_k}(j)$. Now the pigeonhole principle implies there exists some $J' \in D^{k-1}$ with $R(J, j) = 1$ and $q_C(J, j) \geq p_{S_k}(j)/(2|D|^{k-1}) \geq p_{S_k}(j)/r$. By consistency of marginals this certainly implies $p_{S_{t'}}(J_{t'}) \geq p_{S_t}(j)/r \geq \theta$ for all $t' \in [k-1]$. Now for all $t' \in [k-1]$ we know that $J'_{S_{t'}} \in B_{t'}$. Therefore, if we let $J = (J', j)$ we have that $J \in \mathcal{J}$ and $J_k = j$. $\square$

12.3.1 Lattice CSPs: better quantitative dependence on ϵ

As discussed at the end of Section 12.1.1, one cannot hope to improve the approximation guarantee of $1 - O(1/\log(1/\epsilon))$ given by our LP-rounding algorithm, even in the case of Horn-3Sat. On the other hand, for Horn-2Sat it is known [135] that on $(1-\epsilon)$-satisfiable instances one can efficiently find $(1 - O(\epsilon))$-satisfying assignments (indeed, $(1 - 2\epsilon)$-satisfying [113]). One might ask what the algebraic difference is between Horn-2Sat and Horn-3Sat. A notable difference is that the former is a *lattice* CSP.

Subclasses of width-1: lattice and semilattice CSPs. A broad natural subclass of the width-1 CSPs is the class of *semilattice* CSPs. These are CSPs which have a *semilattice polymorphism*, meaning a binary polymorphism $\wedge$ which is associative, commutative, and idempotent. Horn-Sat CSPs are not just width-1 but are in fact semilattice; thus we cannot hope for improved dependence on ϵ even for semilattice CSPs.[2]

An even further subclass is that of *lattice* CSPs. These are CSPs whose relations are preserved by *two* semilattice operations $\wedge$ and $\vee$ which additionally satisfy the "absorption" identity: $\vee(x, \wedge(x, y)) = \wedge(x, \vee(x, y)) = x$. Note that $\vee$ and $\wedge$ extend naturally to polymorphisms of every arity. Good examples of lattice CSPs are "lattice retraction problems". Here there is a fixed lattice poset L; the CSP's domain is L and its constraints are the poset constraint "$\leq$" along with all unary constraints "$=_a$" for $a \in L$.

Robust decidability for lattice CSPs. In this subsection we prove a variant of our Theorem 12.3.1 which shows an efficient LP-based algorithm for finding $(1-O(\epsilon))$-satisfying assignments to $(1 - \epsilon)$-satisfiable *lattice* CSP instances.

We first describe the characterization of lattice CSPs we need. Carvalho, Dalmau, and Krokhin [58] have observed that if $CSP(\Gamma)$ has lattice polymorphisms then it is preserved by what they call an *absorptive block-symmetric operation*. This is an operation f which takes as input tuples (of any positive length) of nonempty subsets of D, outputs an element of D, and has the following properties:

- (Block-symmetry.) $f(B_1, \dots, B_\ell)$ only depends on $\{B_1, \dots, B_\ell\}$.
- (Absorption.) If $B \supseteq B_1$ then $f(B, B_1, \dots, B_\ell) = f(B_1, \dots, B_\ell)$.
- (Preservation.) Let R be an arity-k relation Γ and let $(B_i^j)_{i=1\dots\ell}^{j=1\dots k}$ be nonempty subsets of D. Assume that for each $i \in [\ell]$ there is a $\mathcal{J}_i \subseteq_R B_i^1 \times \cdots \times B_i^k$ all of whose members satisfy R. Then $R(f(B_1^1, \dots, B_\ell^1), \dots, f(B_1^k, \dots, B_\ell^k))) = 1$.

[2]There are CSPs which are width-1 but not semilattice; e.g., the CSP over domain $\{a, b, c, d\}$ with all unary relations and also the binary relations (a, b), (b, a), (c, a), (c, b), (c, d), (d, c), and (d, d).

Indeed, the operation f is simply $f(B_1, \dots, B_\ell) = \bigvee\{\wedge B_i : i \in [\ell]\}$.

We now show:

Theorem 12.3.2. *Let* Γ *be a finite set of relations over the finite domain* D*, each relation having arity at most* K*. Assume that* $CSP(\Gamma)$ *has lattice polymorphisms. Then there is a* $\mathrm{poly}(n, m)$*-time algorithm for* $CSP(\Gamma)$ *which when given an input* $\mathcal{I}$ *with* $\mathsf{val}^{\mathrm{LP}}(\mathcal{I}) = 1 - \epsilon$ *outputs an assignment* $\alpha : V \to D$ *with* $\mathsf{val}_{\mathcal{I}}(\alpha) \geq 1 - O(K2^{|D|})\epsilon$*.*

Proof. As in Theorem 12.3.1, the first step of the algorithm is to solve the LP relaxation of the instance, determining an optimal solution $\{p_v : v \in V\}$, $\{q_{C_i} : i \in [m]\}$ which obtains $\mathsf{val}^{\mathrm{LP}}(\mathcal{I}) = 1 - \epsilon$. Since $CSP(\Gamma)$ has lattice polymorphisms, it has some absorptive block-symmetric operation f. We next describe a randomized LP-rounding algorithm:

1. Set $r = (2K2^{|D|}m)^{-1}$ and choose $\theta \in \{1/r, 2/r, 3/r, \dots, 1\}$ uniformly at random.
2. For each $v \in V$, define $\mathcal{B}_v = \{B \subseteq D : p_v(B) \geq \theta\}$, a nonempty family of nonempty sets. (Here we introduce the notation $p_v(B) = \sum_{b \in B} p_v(b)$.)
3. Output the assignment $\alpha : V \to D$ defined by $\alpha(v) = f(\mathcal{B}_v)$.

We will show for each constraint $C_i = (S_i, R_i)$ that

$$\Pr[R_i(\alpha(S_i)) = 0] \leq K2^{|D|}(\epsilon_i + 1/r) = K2^{|D|}\epsilon_i + 1/2m, \tag{12.5}$$

where $\epsilon_i = \sum_{J : R_i(J)=0} q_{C_i}(J)$ as in the previous proof. It then follows from linearity of expectation that the expected fraction of constraints not satisfied by α is at most $K2^{|D|} \operatorname{avg}\{\epsilon_i\} + 1/2m = K2^{|D|}\epsilon + 1/2m$. We can therefore efficiently deterministically find an α with value at least $1 - K2^{|D|}\epsilon - 1/2m$ by trying all $O(m)$ possible values for θ. This is sufficient to prove the theorem: if $\epsilon < (2K2^{|D|}m)^{-1}$ then α's value exceeds $1 - 1/m$ and hence is in fact 1; if $\epsilon \geq (2K2^{|D|}m)^{-1}$ then the $O(\cdot)$ in the theorem statement covers the loss of $1/2m$.

We now give the analysis justifying (12.5) for each fixed constraint $C_i = (S_i, R_i)$. For simplicity we henceforth write $C = C_i$, $S = S_i$, $R = R_i$ and suppose that R has arity $k \leq K$. It suffices to show that $R(\alpha(S)) = 1$ holds assuming

$$\theta \notin (p_{S_t}(B), p_{S_t}(B) + \epsilon_i] \quad \forall t \in [k],\ \forall B \subseteq D. \tag{12.6}$$

The reason is that the probability of (12.6) not holding is at most $K2^{|D|}(\epsilon_i + 1/r)$. Note that with assumption (12.6), whenever we have $p_{S_t}(B) \geq \theta - \epsilon_i$ it follows that in fact $p_{S_t}(B) \geq \theta$ and thus $B \in \mathcal{B}_{S_t}$.

Claim 12.3.3. *For all $t \in [k]$ and $B \in \mathcal{B}_{S_t}$, there exist $B_1, \ldots, B_k$ with $B_u \in \mathcal{B}_{S_u}$ such that: a) $B_t \subseteq B$; b) there exists $\mathcal{J} \subseteq_S B_1 \times \cdots \times B_k$ with $R(J) = 1$ for all $J \in \mathcal{J}$.*

Proof. Suppose $B \in \mathcal{B}_{S_t}$, so $p_{S_t}(B) \geq \theta$. Letting $\mathcal{J}' = \{J \in D^k : J_t \in B\}$, it follows from consistency of marginals that $q_C(\mathcal{J}') \geq \theta$. Thus if $\mathcal{J}$ is the subset of $\mathcal{J}'$ for which R holds, it follows that $q_C(\mathcal{J}) \geq \theta - \epsilon_i$. For $u \in [k]$, we define $B_u = \{J_u : J \in \mathcal{J}\}$. Certainly $B_t \subseteq B$, and by consistency of marginals we obtain from $q_C(\mathcal{J}) \geq \theta - \epsilon_i$ that $p_{S_u}(B_u) \geq \theta - \epsilon_i$ for each $u \in [k]$. Thus it follows from assumption (12.6) that $B_u \in \mathcal{B}_{S_u}$ for each u, completing the proof of the claim. $\square$

For each choice of $t \in [k]$ and $B \in \mathcal{B}_{S_t}$, take the (names of the) sets $B_1, \ldots, B_k$ given by the above claim and arrange them in a height-k column. Adjoin all of these columns to form a $k \times \ell$ matrix M, where $\ell = \sum_{t=1}^{k} |\mathcal{B}_{S_t}|$. The matrix M has the following properties: (i) each entry in row u is a set in $\mathcal{B}_{S_u}$; (ii) for each set $B \in \mathcal{B}_{S_u}$, some subset of it appears in the u^{th} row of M; (iii) for each column $(B_1, \ldots, B_k)$ of M there is a $\mathcal{J} \subseteq_S B_1 \times \cdots \times B_k$ all of whose members satisfy R.

Suppose we now apply the absorptive block-symmetric operation f to the rows of M, with the u^{th} row producing $j_u \in D$. By (iii), $R(j_1, \ldots, j_k) = 1$. Thus the justification of (12.5) is complete if we can show $j_u = f(\mathcal{B}_{S_u}) = \alpha(S_u)$. But this follows from (i), (ii), and the absorptive property of f. $\square$

Chapter 13

An algorithm for ROBUSTMAXBISECTION

13.1 Introduction

In the MAXCUT problem, we are given a graph and the goal is to partition the vertices into two parts so that a maximum number of edges cross the cut. As one of the most basic problems in the class of constraint satisfaction problems, the study of MAXCUT has been highly influential in advancing the subject of approximability of optimization problems, from both the algorithmic and hardness sides. The celebrated Goemans-Williamson (GW) algorithm for MAXCUT [94] was the starting point for the immense and highly successful body of work on semidefinite programming (SDP) based approximation algorithms. This algorithm guarantees finding a cut whose value (i.e., fraction of edges crossing it) is at least 0.878 times the value of the maximum cut. On graphs that are "almost-bipartite" and admit a partition such that most edges cross the cut, the algorithm performs much better — in particular, if there is a cut such that a fraction $(1-\epsilon)$ of edges cross the cut, then the algorithm finds a partition cutting at least a fraction $1-O(\sqrt{\epsilon})$ of edges. (All through the discussion in the paper, think of ϵ as a very small positive constant.)

On the hardness side, the best known NP-hardness result shows hardness of approximating MAXCUT within a factor greater than $16/17$ [116, 219]. Much stronger and in fact tight inapproximability results are now known conditioned on the Unique Games Conjecture of Khot [136]. In fact, one of the original motivations and applications for the formulation of the UGC in [136] was to show that finding a cut of value larger than $1-o(\sqrt{\epsilon})$ in a graph with Max-Cut value $(1-\epsilon)$ (i.e., improving upon the above-mentioned performance guarantee of the GW algorithm substantially) is likely to be hard. This result was strengthened in [141] to the optimal $1 - O(\sqrt{\epsilon})$ bound, and this paper also showed that the 0.878 approximation factor is the best possible under the UGC. (These results relied, in addition to the UGC, on the Majority is Stablest conjecture, which was proved shortly afterwards in [174].)

There are many other works on MAXCUT, including algorithms that improve on the GW algorithm for certain ranges of the optimum value and integrality gap constructions showing limitations of the SDP based approach. This long line of work on MAXCUT culminated in the paper [183] which obtained the precise integrality gap and approximation threshold curve as a function of the optimum cut value.

Maximum Bisection. Let us consider a closely related problem called MAXBISECTION, which is MAXCUT with a global "balanced cut" condition. In the MAXBISECTION problem, given as input a graph with an even number of vertices, the goal is to partition the vertices into two equal parts while maximizing the fraction of cut edges. Despite the close relation to MAXCUT, the global constraint in MAXBISECTION changes its character substantially, and the known algorithms for approximating MAXBISECTION have weaker guarantees. While MAXCUT has a factor 0.878 approximation algorithm [94], the best known approximation factor for MAXBISECTION equals 0.7027 [91][1], improving on previous bounds of 0.6514 [93], 0.699 [226], and 0.7016 [115].

In terms of inapproximability results, it is known that MAXBISECTION cannot be approximated to a factor larger than $15/16$ unless $\mathrm{NP} \subseteq \bigcap_{\gamma>0} \mathrm{TIME}(2^{n^\gamma})$ [120]. Note that this hardness factor is slightly better than the inapproximability factor of $16/17$ known for MAXCUT [116, 219]. A simple approximation preserving reduction from MAXCUT shows that MAXBISECTION is no easier to approximate than MAXCUT (the reduction is simply to take two disjoint copies of the MAXCUT instance). Therefore, the factor 0.878 Unique-Games hardness for MAXCUT [141] also applies for MAXBISECTION. Further, given a graph that has a bisection cutting $1 - \epsilon$ of the edges, it is Unique-Games hard to find a bisection (or even any partition in fact) cutting $1 - O(\sqrt{\epsilon})$ of the edges.

An intriguing question is whether MAXBISECTION is in fact harder to approximate than MAXCUT (so the global condition really changes the complexity of the problem), or whether there are algorithms for MAXBISECTION that match (or at least approach) what

[1] At the time when this chapter was published. This ratio was later improved by [195] and subsequently [28], as we will mention soon.

is known for MAXCUT.[2] None of the previously known algorithms[3] for MAXBISECTION [93, 226, 115, 91] are guaranteed to find a bisection cutting most of the edges even when the graph has a near-perfect bisection cutting $(1 - \epsilon)$ of the edges (in particular, they may not even cut 75% of the edges). These algorithms are based on rounding a vector solution of a semidefinite programming relaxation into a cut (for example by a random hyperplane cut) and then balancing the cut by moving low-degree vertices from the larger side to the smaller side. After the first step, most of the edges are cut, but the latter rebalancing step results in a significant loss in the number of edges cut. In fact, as we will illustrate with a simple example in Section 13.2, the standard SDP for MAXBISECTION has a large integrality gap: the SDP optimum could be 1 whereas every bisection might only cut less than 0.95 fraction of the edges.

Thus an interesting "qualitative" question is one can one efficiently find an *almost-complete* bisection when promised that one exists. Formally, we ask the following question.

Question 13.1.1. *Is there a polynomial time algorithm that given a graph $G = (V, E)$ with a* MAXBISECTION *solution of value $(1 - \epsilon)$, finds a bisection of value $(1 - g(\epsilon))$, where $g(\epsilon) \to 0$ as $\epsilon \to 0$?*

Indeed, this question motivates the problem of ROBUSTMAXBISECTION defined previously in Definition 2.2.6. We restate the definition here for reader's convenience.

Definition 13.1.2 (ROBUSTMAXBISECTION, re-statement of Definition 2.2.6)**.** *We say that an algorithm A solves the* ROBUSTMAXBISECTION *problem if there exists a function $r : [0, 1] \to [0, 1]$ satisfying $r(\epsilon) \to 0$ as $\epsilon \to 0^+$ such that whenever A is given an undirected graph G with* MAXBISECTION *optimum at least $1 - \epsilon$, A outputs a bisection with $(1 - r(\epsilon))$ of the edges across the bisection.*

Note that without the bisection constraint, we can achieve $g(\epsilon) = O(\sqrt{\epsilon})$, and when $\epsilon = 0$, we can find a bisection cutting all the edges (see the first paragraph of Section 13.2 for details). Thus this question highlights the role of both the global constraint and the "noise" (i.e., ϵ fraction of edges need to be removed to make the input graph bipartite) on the complexity of the problem.

[2]Note that for the problem of minimizing the number of edges cut, the global condition does make a big difference: MINCUT is polynomial-time solvable whereas MINBISECTION is NP-hard.

[3]Also by the time when this chapter was published.

13.1.1 Our contributions

In this chapter, we answer the above question in the affirmative, by proving the following theorem.

Theorem 13.1.3. *There is a randomized polynomial-time algorithm such that for every $\epsilon > 0$, given an edge-weighted graph G with a* MAXBISECTION *solution of value*[4] $(1-\epsilon)$, *finds a* MAXBISECTION *of value* $(1 - O(\sqrt[3]{\epsilon}\log(1/\epsilon)))$.

We remark that for regular graphs any cut with most of the edges crossing it must be near-balanced, and hence we can solve MAXBISECTION by simply reducing to MAXCUT. Thus the interesting instances for our algorithm are non-regular graphs.

Our algorithms are not restricted to finding exact bisections. If the graph has a β-*balanced* cut (which means the two sides have a fraction β and $1-\beta$ of the vertices) of value $(1-\epsilon)$, then the algorithm can find a β-balanced cut of value $1 - O(\epsilon^{1/3}\log(1/\epsilon))$.

Our results are not aimed at improving the general approximation ratio for MAXBISECTION which remains at ≈ 0.7027 [91]. More generally, our work highlights the challenge of understanding the complexity of solving constraint satisfaction problems with global constraints. Algorithmically, the challenge is to ensure that the global constraint is met without hurting the number of satisfied constraints. From the hardness side, the Unique Games based reductions which have led to a complete understanding of the approximation threshold of CSPs [189] are unable to exploit the global constraint to yield stronger hardness results.

13.1.2 Later development

Soon after our work was published, Raghavendra and Tan [195] also studied the ROBUSTMAXBISECTION problem, improving our algorithm and giving an algorithm that finds a MAXBISECTION of value $(1 - O(\sqrt{\epsilon}))$ when there is a MAXBISECTION of value $(1-\epsilon)$. As mentioned earlier, as a function of ϵ, this approximation is best possible assuming the Unique Games Conjecture. However, one thing not very satisfactory about the Raghavendra–Tan algorithm is that their algorithm runs in time $n^{1/\epsilon^{\Theta(1)}}$, which is not completely polynomial in n when ϵ is subconstant (say, $\frac{1}{\mathrm{polylog}(n)}$). It is an interesting open question whether the same approximation guarantee can be obtained in $\mathrm{poly}(n)$ time for every ϵ.

[4]The value of a cut in an edge-weighted graph is defined as the weight of the edges crossing the cut divided by the total weight of all edges.

Using the same technique, Raghavendra and Tan [195] also gave an algorithm substantially improving the approximation ratio of MAXBISECTION to $\approx .852$. This approximation ratio was later improved by [28] to $\approx .8776$. It remains an interesting question how much this can be improved and whether one approach (or even match) the 0.878 factor possible for MAXCUT.

13.2 Method overview

13.2.1 Integrality gap

We begin by describing why the standard SDP for MAXBISECTION has a large gap. Given a graph $G = (V, E)$, this SDP, which is the basis of all previous algorithms for MAXBISECTION starting with that of Frieze and Jerrum [93], solves for unit vectors v_i for each vertex $i \in V$ subject to $\sum_i v_i = 0$, while maximizing the objective function $\mathbf{E}_{e=(i,j)\in E} \frac{1}{4}\|v_i - v_j\|^2$.

This SDP could have a value of 1 and yet the graph may not have any bisection of value more than 0.95 (in particular the optimum is bounded away from 1), as the following example shows. Take G to be the union of *three* disjoint copies of $K_{2m,m}$ (the complete $2m \times m$ bipartite graph) for some even m. It can be seen that every bisection fails to cut at least $m^2/2$ edges, and thus has value at most $11/12$. On the other hand, the SDP has a solution of value 1. Let $\omega = e^{2\pi i/3}$ be the primitive cube root of unity. In the two-dimensional complex plane, we assign the vector/complex number ω^{i-1} (resp. $-\omega^{i-1}$) to all vertices in the larger part (resp. smaller part) of the i^{th} copy of $K_{2m,m}$ for $i = 1, 2, 3$. These vectors sum up to 0 and for each edge, the vectors associated with its endpoints are antipodal.

For all CSPs, a tight connection between integrality gaps (for a certain "canonical" SDP) and inapproximability results is now established [189]. The above gap instance suggests that the picture is more subtle for CSPs with global constraints — in this work we give an algorithm that does much better than the integrality gap for the "basic" SDP. Could a stronger SDP relaxation capture the complexity of approximating CSPs with global constraints such as MAXBISECTION? It is worth remarking that we do not know whether an integrality gap instance of the above form (i.e., $1-\epsilon$ SDP optimum vs. say 0.9 MAXBISECTION value) exists even for the basic SDP augmented with triangle inequalities.

13.2.2 Notations

Suppose we are given a graph $G = (V, E)$. We use the following notation: $E(U) = \{(u, v) \in E : u, v \in U\}$ denotes the set of edges within a set of vertices U, $\mathrm{edges}(U_1, U_2) = \{(u, v) \in E : u \in U_1, v \in U_2\}$ denotes the set of edges between two sets of vertices U_1 and U_2, and $G[U]$ denotes the subgraph of G induced by the set U.

Definition 13.2.1 (Value and bias of cuts)**.** *For a cut $(S, V \setminus S)$ of a graph $G = (V, E)$, we define its* value *to be $\frac{|\mathrm{edges}(S, V\setminus S)|}{|E|}$ (i.e., the fraction of edges which cross the cut) if G is unweighted, and $\frac{w(\mathrm{edges}(S,V\setminus S))}{w(E)}$ if G is edge-weighted with weight function $w : E \to \mathbb{R}^{\geq 0}$ (where for $F \subseteq E$, $w(F) = \sum_{e\in F} w(e)$).*
We define the bias *$\beta \in [0, 1]$ of a cut $(S, V \setminus S)$ to be $\beta = \frac{1}{|V|} \cdot \Big||S| - |V \setminus S|\Big|$, and we say that the cut $(S, V \setminus S)$ is β-biased. (Note that a 0-biased cut is a bisection.)*

Recall that the normalized Laplacian of G is a matrix $\mathcal{L}_G$ whose rows and columns correspond to vertices of G that is defined as follows

$$\mathcal{L}_G(u, v) = \begin{cases} 1, & \text{if } u = v \text{ and } d_u \neq 0, \\ -1/\sqrt{d_u d_v}, & \text{if } (u, v) \in E, \\ 0, & \text{otherwise,} \end{cases}$$

where d_u is the degree of the vertex u. Let $\lambda_2(\mathcal{L}_G)$ be the second smallest eigenvalue of $\mathcal{L}_G$. We abuse the notation by letting $\lambda_2(G) = \lambda_2(\mathcal{L}_G)$. We define the volume of a set $U \subseteq V$ as $\mathrm{vol}(U) = \mathrm{vol}_G(U) = \sum_{u\in U} d_u$.

We will use the following version of Cheeger's inequality.

Theorem 13.2.2 (Cheeger's inequality for non-regular graphs [67])**.** *For every graph $G = (V, E)$,*

$$\lambda_2(G)/2 \leq \phi(G) \leq \sqrt{2\lambda_2(G)},$$

where $\phi(G)$ is the expansion of G,

$$\phi(G) \equiv \min_{S\subseteq V} \frac{|\mathrm{edges}(S, V \setminus S)|}{\min(\mathrm{vol}(S), \mathrm{vol}(V \setminus S))}.$$

Moreover, we can efficiently find a set $A \subseteq V$ such that $\mathrm{vol}(A) \leq \mathrm{vol}(V)/2$ and $|\mathrm{edges}(A, V \setminus A)|/\mathrm{vol}(A) \leq \sqrt{2\lambda_2(G)}$.

For any two disjoint sets $X, Y \subseteq V$, let $\mathrm{uncut}(X, Y) = |E(X) + E(Y)|/|E(X \cup Y)|$ be the fraction of edges of $G[X \cup Y]$ that do not cross the cut (X, Y). We say that a cut (X, Y) of V is perfect if $\mathrm{uncut}(X, Y) = 0$.

13.2.3 Our approach

In this section, we give a brief overview of our algorithm. It is instructive to consider first the case when G has a perfect bisection cut. In this case, G is a bipartite graph. If G has only one connected component, each part of this component has the same number of vertices, so this is the desired bisection. Now assume that G has several connected components. Then each connected component C of G is a bipartite graph with two parts X_C and Y_C. Since all edges are cut in the optimal solution, X_C must lie on one side of the optimal cut and Y_C on the other. So in order to find a perfect bisection (X, Y), for every connected component C we need to either (i) add X_C to X and Y_C to Y or (ii) add X_C to Y and Y_C to X so that $|X| = |Y| = |V|/2$. We can do that using dynamic programming.

Our algorithm for almost satisfiable instances proceeds in a similar way. Assume that the optimal bisection cuts a $(1 - \epsilon)$ fraction of edges.

1. In a preprocessing step, we use the algorithm of Goemans and Williamson [94] to find an approximate maximum cut in G. A fraction $1 - O(\sqrt{\epsilon})$ of edges cross this cut. We remove all uncut edges and obtain a bipartite graph. We denote the parts of this graph by A and B. (Of course, in general $|A| \neq |B|$.)

2. Then we recursively partition G into pieces $W_1, \ldots, W_s$ using Cheeger's Inequality (see Lemma 13.3.1). Every piece is either a sufficiently small subgraph, which contains at most an ϵ fraction of all vertices, or is a spectral expander, with $\lambda_2 \geq \epsilon^{2/3}$. There are very few edges between different pieces, so we can ignore them later. In this step, we obtain a collection of induced subgraphs $G[W_1], \ldots, G[W_s]$ with very few edges going between different subgraphs.

3. Now our goal is to find an "almost perfect" cut in every $G[W_i]$, then combine these cuts and get a bisection of G. Note that every $G[W_i]$ is bipartite and therefore has a perfect cut (since G is bipartite after the preprocessing step). However, we cannot restrict our attention only to this perfect cut since the optimal solution (S, T) can cut $G[W_i]$ in another proportion. Instead, we prepare a list $\mathcal{W}_i$ of "candidate cuts" for each $G[W_i]$ that cut W_i in different proportions. One of them is close to the cut $(W_i \cap S, W_i \cap T)$ (the restriction of the optimal cut to W_i).

4. If $G[W_i]$ is an expander, we find a candidate cut that cuts $G[W_i]$ in a given proportion by moving vertices from one side of the perfect cut $(W_i \cap A, W_i \cap B)$ to the other, greedily (see Lemma 13.4.1 and Lemma 13.4.2).

5. If $G[W_i]$ is small, we find a candidate cut that cuts $G[W_i]$ in a given proportion using semi-definite programming (see Lemma 13.4.3 and Corollary 13.4.4). We solve an

SDP relaxation similar to the Goemans–Williamson relaxation [94] with "ℓ_2^2-triangle inequalities", and then find a cut by using hyperplane or threshold rounding.

In fact, the cut that we find can be more unbalanced than $(W_i \cap S, W_i \cap T)$ but this is not a problem since the set W_i is small. Note however that if a cut of another piece W_j is very unbalanced than we might need to find a cut of W_i that is unbalanced in the other direction. So it is important that the candidate cut of W_i is at least as unbalanced as $(W_i \cap S, W_i \cap T)$.

6. Finally, we combine candidate cuts of subgraphs $G[W_i]$ into one balanced cut of the graph G, in the optimal way, using dynamic programming (see Lemma 13.5.1).

13.2.4 Organization

The rest of the chapter is devoted to the full description and proof of the algorithm. In Section 13.3, we partition the graph into expanders and small pieces, after proper preprocessing. In Section 13.4, we produce a list of candidate cuts for each expander and small piece, by different methods. In Section 13.5, we show how to choose one candidate cut for each part. In Section 13.6, we put everything together to finish the proof of Theorem 13.1.3.

13.3 Preprocessing and partitioning graph G

In this section, we present the preprocessing and partitioning steps of our algorithms. We will assume that we know the value of the optimal solution $\mathsf{opt} = 1 - \epsilon_{\mathsf{opt}}$ (with a high precision). If we do not, we can run the algorithm for many different values of ϵ and output the best of the bisection cuts we find.

13.3.1 Preprocessing: Making G bipartite and unweighted

In this section, we show that we can assume that the graph G is bipartite, with parts A and B, unweighted, and that $|E| \leq O(|V|/\epsilon_{\mathsf{opt}}^2)$.

First, we "sparsify" the edge-weighted graph $G = (V, E)$, and make the graph unweighted: we sample $O(\epsilon_{\mathsf{opt}}^{-2}|V|)$ edges (according to the weight distribution) with replacement from G, then with high probability, every cut has the same cost in the original graph as in the new graph, up to an additive error ϵ_{opt} (by Chernoff's bound). So we assume that $|E| \le O(\epsilon_{\mathsf{opt}}^{-2}|V|)$.

We apply the algorithm of Goemans and Williamson to G and find a partitioning of G into two pieces A and B so that only an $O(\sqrt{\epsilon_{\mathsf{opt}}})$ fraction of edges lies within A or within B.

13.3.2 Partitioning

In this section, we describe how we partition G into pieces.

Lemma 13.3.1. *Given a graph $G = (V, E)$, and parameters $\delta \in (0, 1)$ and $\lambda \in (0, 1)$ such that $|E| = O(|V|/\delta^2)$, we can find a partitioning of V into disjoint sets $U_1, \dots, U_p$ ("small sets"), and $V_1, \dots, V_q$ ("expander graphs"):*

$$V = \bigcup_i U_i \cup \bigcup_j V_j,$$

in polynomial time, so that

1. *$|U_i| \le \delta|V|$ for each $1 \le i \le p$;*
2. *$\lambda_2(G[V_i]) \ge \lambda$ for each $1 \le i \le q$;*
3. *$\sum_i |E(U_i)| + \sum_j |E(V_j)| \ge (1 - O(\sqrt{\lambda}\log(1/\delta)))|E|$.*

Proof. We start with a trivial partitioning $\{V\}$ of V and then iteratively refine it. Initially, all sets in the partitioning are "active"; once a set satisfies conditions 1 or 2 of the lemma, we permanently mark it as "passive" and stop subdividing it. We proceed until all sets are passive. Specifically, we mark a set S as passive in two cases. First, if $|S| \le \delta|V|$ then we add S to the family of sets U_i. Second, if $\lambda_2(G[S]) \ge \lambda$ then we add S to the family of sets V_i.

We subdivide every active S into smaller pieces by applying the following easy corollary of Cheeger's inequality (Theorem 13.2.2) to $H = G[S]$.

Corollary 13.3.2. *Given a graph $H = (S, E(H))$ and a threshold $\lambda > 0$, we can find, in polynomial time, a partition $S_1, S_2, \cdots, S_t$ of S such that*

1. *$|E(S_i)| \le |E(S)|/2$ or $\lambda_2(H[S_i]) \ge \lambda$, for each $1 \le i \le t$.*
2. *$\sum_{i<j} |\mathrm{edges}(S_i, S_j)| \le \sqrt{8\lambda}|E(H)|$.*

3. *each graph $H[S_i]$ is connected.*

Proof. If $\lambda_2(H) \geq \lambda$ then we just output a trivial partition $\{S\}$. Otherwise, we apply Theorem 13.2.2 to $H_1 = H$, find a set S_1 s.t. $\mathrm{vol}_{H_1}(S_1) \leq \mathrm{vol}_{H_1}(S)/2$ and $|\mathrm{edges}(S_1, S \setminus S_1)|/\mathrm{vol}_{H_1}(S_1) \leq \sqrt{2\lambda_2(H_1)} \leq \sqrt{2\lambda}$. Then we remove S_1 from H_1, obtain a graph H_2 and iteratively apply this procedure to H_2. We stop when either $\lambda_2(H_i) \geq \lambda$ or $|E(H_i)| \leq |E(S)|/2$.

We verify that the obtained partitioning $S_1, \dots, S_t$ of S satisfies the first condition. For each $i \in \{1, \dots, t-1\}$, we have $|E(S_i)| \leq \mathrm{vol}_{H_i}(S_i)/2 \leq \mathrm{vol}_{H_i}(V(H_i))/4 = E(H_i)/2 \leq |E(H)|/2$. Our stopping criterion guarantees that S_t satisfies the first condition. We verify the second condition.

$$\sum_{i<j} |\mathrm{edges}(S_i, S_j)| = \sum_{i=1}^{t-1} |\mathrm{edges}(S_i, V(H_i) \setminus S_i)|$$
$$\leq \sum_{i=1}^{t-1} \sqrt{2\lambda}\, \mathrm{vol}_{H_i}(S_i) \leq \sqrt{2\lambda}\, \mathrm{vol}_H(S) = 2\sqrt{2\lambda}|E(H)|.$$

Finally, if for some i, $H[S_i]$ is not connected, we replace S_i in the partitioning with the connected components of $H[S_i]$. $\square$

By the definition, sets U_i and V_j satisfy properties 1 and 2. It remains to verify that

$$\sum_{i=1}^{p} |E(U_i)| + \sum_{j=1}^{q} |E(V_j)| \geq (1 - O(\sqrt{\lambda} \log(1/\delta)))|E|.$$

We first prove that the number of iterations is $O(\log(1/\delta))$. Note that if S is an active set and T is its parent then $|E(S)| \leq |E(T)|/2$. Set V contains $O(|V|/\delta^2)$ edges. Every active set S contains at least $\delta|V|/2$ edges, since $|E(S)| \geq |S| - 1 \geq \delta|V|/2$ (we use that $G[S]$ is connected). Therefore, the number of iterations is $O(\log_2((|V|/\delta^2)/(\delta|V|/2))) = O(\log 1/\delta)$.

We finally observe that when we subdivide a set S, we cut $O(\sqrt{\lambda}|E(S)|)$ edges. At each iteration, since all active sets are disjoint, we cut at most $O(\sqrt{\lambda}|E|)$ edges. Therefore, the total number of edges cut in all iterations is $O(\sqrt{\lambda} \log(1/\delta))|E|$. $\square$

13.4 Finding cuts in sets U_i and V_i

In the previous section, we showed how to partition the graph G into the union of "small graphs" $G[U_i]$ and expander graphs $G[V_i]$. We now show how to find good "candidate cuts" in each of these graphs.

13.4.1 Candidate cuts in V_i

In this section, first we prove that there is essentially only one almost perfect maximum cut in an expander graph (Lemma 13.4.1). That implies that every almost perfect cut in the graph $G[V_i]$ should be close to the perfect cut $(V_i \cap A, V_i \cap B)$. Using that we construct a list of good candidate cuts (Lemma 13.4.2). One of these cuts is close to the restriction of the optimal cut to subgraph $G[V_i]$.

Lemma 13.4.1. *Suppose we are given a graph $H = (V, E)$ and two cuts (S_1, T_1) and (S_2, T_2) of G, each of value at least $(1 - \delta)$. Then*

$$\min\{\mathrm{vol}_H(S_1 \triangle S_2), \mathrm{vol}_H(S_1 \triangle T_2)\} \leq 4\delta|E|/\lambda_2(H).$$

Proof. Let

$$X = S_1 \triangle S_2 = (S_1 \cap T_2) \cup (S_2 \cap T_1);$$
$$Y = S_1 \triangle T_2 = (S_1 \cap T_1) \cup (S_2 \cap T_2).$$

Note that $V = X \cup Y$. There are at most $2\delta|E|$ edges between X and Y, since

$$\mathrm{edges}(X, Y) \subset E(S_1) \cup E(S_2) \cup E(T_1) \cup (T_2),$$

$|E(S_1) \cup E(T_1)| \leq \delta|E|$ and $|E(S_2) \cup E(T_2)| \leq \delta|E|$.

On the other hand, by Cheeger's inequality (Theorem 13.2.2), we have

$$\frac{|\mathrm{edges}(X, Y)|}{\min(\mathrm{vol}_H(X), \mathrm{vol}_H(Y))} \geq \lambda_2(H)/2.$$

Therefore,

$$\min(\mathrm{vol}_H(X), \mathrm{vol}_H(Y)) \leq 2|\mathrm{edges}(X, Y)|/\lambda_2(H) \leq \frac{4\delta|E|}{\lambda_2(H)}.$$

□

Consider one of the sets V_i. Let $H = G[V_i]$. Denote $A_i = V_i \cap A$ and $B_i = V_i \cap B$. We sort all vertices in A_i and B_i w.r.t. their degrees in H. Now we are ready to define the family of candidates cuts $(X_0, Y_0), \dots, (X_{|V_i|}, Y_{|V_i|})$ for $G[V_i]$. For each j, we define (X_j, Y_j) as follows.

- If $j \le |A_i|$ then X_j consists of j vertices of A_i with highest degrees, and Y_j consists of the remaining vertices of H (i.e. Y_j contains all vertices of B_i as well as $|A_i| - j$ lowest degree vertices of A_i).
- If $j \ge |A_i|$ then Y_j consists of $|V_i| - j$ vertices of B_i with highest degrees, and X_j consists of the remaining vertices of H.

Clearly, $|X_j| = j$ and $|Y_j| = |V_i| - j$. Let (S, T) be the restriction of the optimal bisection of G to H. We will show that one of the cuts (X_j, Y_j) is not much worse than (S, T). By Lemma 13.4.1 applied to cuts (A_i, B_i) and (S, T) (note that $\mathrm{uncut}(A_i, B_i) = 0$),

$$\min\{\mathrm{vol}_H(A_i \triangle S), \mathrm{vol}_H(A_i \triangle T)\} \le \frac{4 \cdot \mathrm{uncut}(S,T)|E(H)|}{\lambda_2(H)}.$$

Assume without loss of generality that $\mathrm{vol}_H(A_i \triangle S) \le 4E(H)/\lambda_2(H)$ (otherwise, rename sets X and Y). We show that $\mathrm{vol}_H(A_i \triangle X_{|S|}) \le \mathrm{vol}_H(A_i \triangle S)$. Consider the case $|A_i| \ge |S|$. Note that by the definition of $X_{|S|}$, the set $X_{|S|}$ has the largest volume among all subsets of A_i of size at most $|S|$. Correspondingly, $A_i \setminus X_{|S|}$ has the smallest volume among all subsets of A_i of size at least $|A_i| - |S|$. Finally, note that $|A_i \setminus S| \ge |A_i| - |S|$. Therefore,

$$\mathrm{vol}_H(A_i \triangle X_{|S|}) = \mathrm{vol}_H(A_i \setminus X_{|S|}) \le \mathrm{vol}_H(A_i \setminus S) \le \mathrm{vol}_H(A_i \triangle S).$$

The case when $|A_i| \le |S|$ is similar. We conclude that

$$\mathrm{vol}_H(A_i \triangle X_{|S|}) \le 4 \cdot \mathrm{uncut}(S,T)|E_H|/\lambda_2(H).$$

Therefore, the size of the cut $(X_{|S|}, Y_{|S|})$ is at least

$$|E(H)| - \mathrm{vol}_H(A_i \triangle X_{|S|}) \ge \Big(1 - \frac{4 \cdot \mathrm{uncut}(S,T)}{\lambda_2(H)}\Big)|E(H)|.$$

We have thus proved the following lemma.

Lemma 13.4.2. *There is a polynomial time algorithm that given a graph $H = G([V_i])$ finds a family of cuts $\mathcal{V}_i = \{(X_1, Y_1), \dots, (X_{|V_i|}, Y_{|V_i|})\}$ such that for every cut (S, T) of H there exists a cut $(X, Y) \in \mathcal{V}_i$ with $|X| = \min(|S|, |T|)$ and*

$$\mathrm{uncut}(X, Y) \le \frac{4 \cdot \mathrm{uncut}(S,T)}{\lambda_2(H)}\ .$$

13.4.2 Candidate cuts in U_i

In this section, we show how to find candidate cuts for the small parts, i.e., the induced subgraphs $G[U_i]$.

Lemma 13.4.3. *Suppose we are given a graph $H = (U, E)$ and two parameters $0 \le \theta \le 1/2$ and $0 < \Delta < 1$. Then in polynomial time we can find a cut (X, Y) such that for every cut (S, T) in H, with $|S| \le \theta|U|$, we have*

1. $\mathrm{uncut}(X, Y) \le O(\sqrt{\mathrm{uncut}(S, T)} + \mathrm{uncut}(S, T)/\Delta)$.
2. $|X| \le (\theta + \Delta)|U|$.

Proof. Let (S, T) be the maximum cut among all cuts with $|S| \le t|U|$ (of course, our algorithm does not know (S, T)). Let $\epsilon_H = \mathrm{uncut}(S, T)$. We may assume that our algorithm knows the value of ϵ_H (with high precision) — as otherwise, we can run our algorithm on different values of ϵ and output the best of the cuts the algorithm finds.

We write the following SDP program. For every vertex $i \in U$, we introduce a unit vector v_i. Additionally, we introduce a special unit vector v_0.

$$
\begin{aligned}
&\text{Maximize} && \frac{1}{|U|}\sum_{i\in U}\langle v_0, v_i\rangle && \\
&\text{Subject to} && \frac{1}{4|E|}\sum_{(i,j)\in E}\|v_i + v_j\|^2 \le \epsilon_H && \\
& && \|v_i\|^2 = 1 && \forall i \in V \cup \{0\} \\
& && |\langle v_i + v_j, v_0\rangle| \le \frac{\|v_i + v_j\|^2}{2} && \forall i, j \in V.
\end{aligned}
$$

The "intended solution" to this SDP is $v_i = v_0$ if $i \in T$ and $v_i = -v_0$ if $i \in S$ (vector v_0 is an arbitrary unit vector). Clearly, this solution satisfies all SDP constraints. In particular, it satisfies the last constraint ("an ℓ_2^2-triangle inequality") since the left hand side is positive only when $v_i = v_j$, then $|\langle v_i + v_j, v_0\rangle| = \frac{\|v_i+v_j\|^2}{2} = 2$. The value of this solution is $(|T| - |S|)/|U| \ge 1 - 2\theta$.

We solve the SDP and find the optimal SDP solution $\{v_i\}$. Note that $\sum_{i\in U}\langle v_0, v_i\rangle \ge (1 - 2\theta)|U|$.

Let $\Delta' = 2\Delta/3$. Choose $r \in [\Delta', 2\Delta']$ uniformly at random. Define a partition of U into sets Z_k, $0 \le k < 1/\Delta'$, as follows: let $Z_k = \{i : k\Delta' + r < |\langle v_0, v_i\rangle| \le (k+1)\Delta' + r\}$

for $k \geq 1$ and $Z_0 = \{i : -\Delta' - r \leq \langle v_0, v_i \rangle \leq \Delta' + r\}$. We bound the probability that the endpoints of an edge (i, j) belong to different sets Z_k. Note that if no point from the set $\{\pm(k\Delta' + r) : k \geq 1\}$ lies between $|\langle v_i, v_0 \rangle|$ and $|\langle v_j, v_0 \rangle|$ then i and j belong to the same set Z_k. The distance between $|\langle v_i, v_0 \rangle|$ and $|\langle v_j, v_0 \rangle|$ is at most $|\langle v_i + v_j, v_0 \rangle|$. Therefore, the probability (over r) that i and j belong to different sets Z_k is at most $|\langle v_i + v_j, v_0 \rangle|/\Delta'$. So the expected number of cut edges is at most

$$\frac{1}{\Delta'} \sum_{(i,j)\in E} |\langle v_i + v_j, v_0 \rangle| \leq \frac{1}{2\Delta'} \sum_{(i,j)\in E} \|v_i + v_j\|^2 \leq \frac{2|E|\epsilon_H}{\Delta'}. \tag{13.1}$$

For each $k \geq 1$, let $Z_k^+ = \{i \in Z_k \mid \langle v_i, v_0 \rangle > 0\}$ and $Z_k^- = \{i \in Z_k \mid \langle v_i, v_0 \rangle < 0\}$. We use hyperplane rounding of Goemans and Williamson [94] to divide Z_0 into two sets Z_0^+ and Z_0^-. We are ready to define sets X and Y. For each k, we add vertices from the smaller of the two sets Z_k^+ and Z_k^- to X, and vertices from the larger of them to Y.

Now we bound $\mathrm{uncut}(X, Y)$. Note that

$$|\mathrm{uncut}(X, Y)| \leq \sum_{k<l} |\mathrm{edges}(Z_k, Z_l)| + \sum_{k\geq 0} (|E(Z_k^+)| + |E(Z_k^-)|).$$

We have already shown that $\sum_{k<l} |\mathrm{edges}(Z_k, Z_l)|$ is less than $2\epsilon_H|E|/\Delta'$ in expectation. If $(i, j) \in E(Z_k^+)$ or $(i, j) \in E(Z_k^-)$ for $k \geq 1$ then $|\langle v_i + v_j, v_0 \rangle| \geq \Delta'$. Therefore,

$$\sum_{k\geq 1} (|E(Z_k^+)| + |E(Z_k^-)|) \leq \frac{2\epsilon_H|E|}{\Delta'} = \frac{3\epsilon_H|E|}{\Delta} .$$

Finally, note that when we divide Z_0, the fraction of edges of $E(Z_0)$ that do not cross the random hyperplane is $O(\sqrt{\epsilon_0})$ (in expectation) where

$$\epsilon_0 = \frac{1}{4|E(Z_0)|} \sum_{(i,j)\in E(Z_0)} \|v_i + v_j\|^2 \leq \frac{1}{4|E(Z_0)|} \sum_{(i,j)\in E} \|v_i + v_j\|^2 \leq \frac{\epsilon_H \cdot |E|}{|E(Z_0)|}.$$

Thus,

$$\mathbf{E}\big[|E(Z_0^+)| + |E(Z_0^-)|\big|r\big] \leq O\left(\sqrt{\epsilon_H|E|/|E(Z_0)|}\right) |E(Z_0)| \leq O(\sqrt{\epsilon_H})|E|.$$

Combining the above upper bounds, we conclude that

$$\mathbf{E}[\mathrm{uncut}(X, Y)] \leq O\left(\frac{\epsilon_H}{\Delta} + \sqrt{\epsilon_H}\right) .$$

Finally, we estimate the size of the set X. Note that if $v_i \in Z_k^+$ then $|\langle v_i, v_0\rangle - k\Delta'| \le 3\Delta'$, if $v_i \in Z_k^-$ then $|\langle v_i, v_0\rangle + k\Delta'| \le 3\Delta'$. Therefore, $\sum_{i\in Z_k}\langle v_i, v_0\rangle \le k(|Z_k^+| - |Z_k^-|)\Delta' + 3\Delta'|Z_k|$, which implies

$$\sum_k k\big(|Z_k^+| - |Z_k^-|\big)\Delta' \ge \sum_{i\in U}\langle v_i, v_0\rangle - 3\Delta'|U| \ge (1-2\theta-3\Delta')|U| \,.$$

Therefore,

$$\begin{aligned}
|Y| - |X| &= \sum_k \big||Z_k^+| - |Z_k^-|\big| \ge \sum_{k:|Z_k^+|-|Z_k^-|>0} \big(|Z_k^+| - |Z_k^-|\big) \\
&\ge \sum_{k:|Z_k^+|-|Z_k^-|>0} (k\Delta')\big(|Z_k^+| - |Z_k^-|\big) \\
&\ge \sum_k k\big(|Z_k^+| - |Z_k^-|\big)\Delta' \ge (1-2\theta-3\Delta')|U| \,,
\end{aligned}$$

implying $|X| \le (\theta + 3\Delta'/2)|U| = (\theta+\Delta)|U|$. □

We apply this algorithm to every graph $G[U_i]$ and every $\theta = k/|U_i|$, $0 < k \le |U_i|/2$, and obtain a list of candidate cuts. We get the following corollary.

Corollary 13.4.4. *There is a polynomial time algorithm that given a graph $H = G([U_i])$ and a parameter $\Delta \in (0,1)$ finds a family of cuts $\mathcal{U}_i$ such that for every cut (S,T) of H there exists a cut $(X,Y) \in \mathcal{U}_i$ with $|X| \le \min(|S|,|T|) + \Delta|U_i|$ and*

$$\mathrm{uncut}(X,Y) \le O\left(\sqrt{\mathrm{uncut}(S,T)} + \frac{\mathrm{uncut}(S,T)}{\Delta}\right).$$

13.5 Combining candidate cuts

In this section, we show how to choose one candidate cut for each set U_i and V_j.

For brevity, we denote $W_i = U_i$ for $i \in \{1,\dots,p\}$ and $W_{p+j} = V_j$ for $j \in \{1,\dots,q\}$. Similarly, $\mathcal{W}_i = \mathcal{U}_i$ for $i \in \{1,\dots,p\}$ and $\mathcal{W}_{p+j} = \mathcal{V}_j$ for $j \in \{1,\dots,q\}$ Then $W_1,\dots,W_{p+q}$ is a partitioning of V, and $\mathcal{W}_i$ is a family of cuts of $G[W_i]$.

We say that a cut (X,Y) of G is a combination of candidate cuts from $\mathcal{W}_i$ if the restriction of (X,Y) to each W_i belongs to $\mathcal{W}_i$ (we identify cuts (S,T) and (T,S)).

Lemma 13.5.1. *There exists a polynomial time algorithm that given a graph $G = (V, E)$ and a threshold $\zeta \in [0, 1/2]$, sets W_i and families of cuts $\mathcal{W}_i$, finds the maximum cut among all combination cuts (X, Y) with $|X|, |Y| \in [(1/2 - \zeta)|V|, (1/2 + \zeta)|V|]$.*

Proof. We solve the problem by dynamic programming. Denote $H_k = G[\bigcup_{i=1}^{k} W_i]$. For every $a \in \{1, \ldots, p+q\}$ and $b \in \{1, \ldots, |G[H_a]|\}$, let $Q[a, b]$ be the size of the maximum cut among all combination cuts (X, Y) on H_a with $|X| = b$ ($Q[a, b]$ equals $-\infty$ if there are no such cuts). We loop over all value of a from 1 to $p + q$ and fill out the table Q using the following formula

$$Q[a, b] = \max_{(X,Y)\in\mathcal{W}_a \text{ or } (Y,X)\in\mathcal{W}_a} (Q[a - 1, b - |X|] + |\mathrm{edges}(X, Y)|),$$

where we assume that $Q[0, 0] = 0$, and $Q[a, b] = -\infty$ if $a \leq 0$ and $b \leq 0$ and $(a, b) \neq (0, 0)$.

Finally the algorithm outputs maximum among $T[p + q, \lceil (1/2 - \zeta)|V| \rceil], \ldots, T[p + q, \lfloor (1/2 + \zeta)|V| \rfloor]$, and the corresponding combination cut. □

Finally, we prove that there exists a good almost balanced combination cut.

Lemma 13.5.2. *Let $G = (V, E)$ be a graph. Let $V = \bigcup_i U_i \cup \bigcup_j V_j$ be a partitioning of V that satisfies conditions of Lemma 13.3.1, and $\mathcal{U}_i$ and $\mathcal{V}_j$ be families of candidate cuts that satisfy conditions of Corollary 13.4.4 and Lemma 13.4.2, respectively. Then there exists a composition cut (X, Y) such that*

$$\left| \frac{|X|}{|V|} - \frac{1}{2} \right| \leq \max(\Delta, \delta)$$

and

$$\mathrm{uncut}(X, Y) \leq O\left(\sqrt{\lambda} \log(1/\delta) + \sqrt{\mathrm{uncut}(S_{\mathsf{opt}}, T_{\mathsf{opt}})} + \mathrm{uncut}(S_{\mathsf{opt}}, T_{\mathsf{opt}}) \left(\frac{1}{\lambda} + \frac{1}{\Delta} \right) \right),$$

where $(S_{\mathsf{opt}}, T_{\mathsf{opt}})$ is the optimal bisection of G.

Proof. Consider the optimal bisection cut $(S_{\mathsf{opt}}, T_{\mathsf{opt}})$. We choose a candidate cut for every set V_i. By Lemma 13.4.2, for every V_i there exists a cut $(X_i, Y_i) \in \mathcal{V}_i$ such that

$$\mathrm{uncut}(X_i, Y_i) \leq 4\mathrm{uncut}(S_{\mathsf{opt}} \cap V_i, T_{\mathsf{opt}} \cap V_i)/\lambda_2(G[V_i]) \leq 4\mathrm{uncut}(S_{\mathsf{opt}} \cap V_i, T_{\mathsf{opt}} \cap V_i)/\lambda, \tag{13.2}$$

and $|X_i| = \min(|S_{\mathsf{opt}} \cap V_i|, |T_{\mathsf{opt}} \cap V_i|)$. We define sets X^V and Y^V as follows. For each i, we add X_i to X^V if $|X_i| = |S_{\mathsf{opt}} \cap V_i|$, and we add Y_i to X^V, otherwise (i.e. if $|Y_i| = |S_{\mathsf{opt}} \cap V_i|$). Similarly, we add Y_i to Y^V if $|Y_i| = |T_{\mathsf{opt}} \cap V_i|$, and we add X_i to Y^V, otherwise. Clearly, (X^V, Y^V) is a candidate cut of $\bigcup_i V_i$ and $|X^V| = |S_{\mathsf{opt}} \cap \bigcup_i V_i|$. Assume without loss of generality that $|X^V| \geq |Y^V|$.

Now we choose a candidate cut for every set U_i. By Corollary 13.4.4, for every U_i there exists a cut $(X_i', Y_i') \in \mathcal{U}_i$ such that

$$\operatorname{uncut}(X_i', Y_i') \leq O\left(\sqrt{\operatorname{uncut}(S_{\mathsf{opt}} \cap U_i, T_{\mathsf{opt}} \cap U_i)} + \frac{\operatorname{uncut}(S_{\mathsf{opt}} \cap U_i, T_{\mathsf{opt}} \cap U_i)}{\Delta}\right), \tag{13.3}$$

and $|X_i'| \leq \min(|S_{\mathsf{opt}} \cap U_i|, |T \cap U_i|) + \Delta |U_i|$. We assume that X_i' is the smaller of the two sets X_i' and Y_i'.

We want to add one of the sets X_i' and Y_i' to X^V, and the other set to Y^V so that the resulting cut (X, Y) is almost balanced. We set $X = X^V$ and $Y = Y^V$. Then consequently for every i from 1 to p, we add X_i' to the larger of the sets X and Y, and add Y_i' to the smaller of the two sets (recall that X_i' is smaller than Y_i'). We obtain a candidate cut (X, Y) of G.

We show that $\big||X|/|V| - 1/2\big| \leq \max(\Delta, \delta)$. Initially, $|X| = |X^V| \geq |Y| = |Y^V|$. If at some point X becomes smaller than Y then after that $\big||X| - |Y|\big| \leq \delta |V|$ since at every step $\big||X| - |Y|\big|$ does not change by more than $|U_i| \leq \delta |V|$. So in this case $\big||X|/|V| - 1/2\big| \leq \delta$. So let us assume that the set X always remains larger than Y. Then we always add X_i' to X and Y_i' to Y. We have

$$\begin{aligned}
|X| &= \Big|X^V \cup \bigcup_i X_i'\Big| \\
&\leq \sum_{j=1}^{q} |S_{\mathsf{opt}} \cap V_j| + \sum_{i=1}^{p} (\min(|S_{\mathsf{opt}} \cap U_i|, |T_{\mathsf{opt}} \cap U_i|) + \Delta |U_i|) \\
&\leq \sum_{j=1}^{q} |S_{\mathsf{opt}} \cap V_j| + \sum_{i=1}^{p} |S_{\mathsf{opt}} \cap U_i| + \Delta |V| \\
&= |S_{\mathsf{opt}}| + \Delta |V| = (1/2 + \Delta)|V|.
\end{aligned}$$

It remains to bound $\operatorname{uncut}(X, Y)$. We have,

$$\operatorname{uncut}(X, Y)|E| \leq \sum_{1 \leq i < j \leq p} |\operatorname{edges}(U_i, U_j)| + \sum_{1 \leq i < j \leq q} |\operatorname{edges}(V_i, V_j)| + \sum_{\substack{1 \leq i \leq p \\ 1 \leq j \leq q}} |\operatorname{edges}(U_i, V_j)|$$

$$+\sum_{1\le i\le p}\mathrm{uncut}(X'_i,Y'_i)|E(U_i)|+\sum_{1\le j\le q}\mathrm{uncut}(X_j,Y_j)|E(V_j)|\,.$$

By Lemma 13.3.1, the sum of the first three terms is at most $O(\sqrt{\lambda}\log(1/\delta))|E|$. From (13.3), we get

$$\begin{aligned}
&\sum_{1\le i\le p}\mathrm{uncut}(X'_i,Y'_i)|E(U_i)|\\
\le&O(1)\sum_{i=1}^{p}\Big(\sqrt{\mathrm{uncut}(S_{\mathsf{opt}}\cap U_i,T_{\mathsf{opt}}\cap U_i)}+\mathrm{uncut}(S_{\mathsf{opt}}\cap U_i,T_{\mathsf{opt}}\cap U_i)/\Delta\Big)|E(U_i)|\\
=&O(1)\sum_{i=1}^{p}\sqrt{\big(|E(S_{\mathsf{opt}}\cap U_i)|+|E(T_{\mathsf{opt}}\cap U_i)|\big)}\cdot\sqrt{|E(U_i)|}\\
&\quad+O(1)\sum_{i=1}^{p}\frac{|E(S_{\mathsf{opt}}\cap U_i)|+|E(T_{\mathsf{opt}}\cap U_i)|}{\Delta}\\
\le&O(1)\sqrt{\sum_{i=1}^{p}\big(|E(S_{\mathsf{opt}}\cap U_i)|+|E(T_{\mathsf{opt}}\cap U_i)|\big)}\cdot\sqrt{\sum_{i=1}^{p}|E(U_i)|}+O\left(\frac{\mathrm{uncut}(S_{\mathsf{opt}},T_{\mathsf{opt}})\cdot|E|}{\Delta}\right)\\
\le&O\Big(\sqrt{\mathrm{uncut}(S_{\mathsf{opt}},T_{\mathsf{opt}})}+\frac{\mathrm{uncut}(S_{\mathsf{opt}},T_{\mathsf{opt}})}{\Delta}\Big)\cdot|E|.
\end{aligned}$$

From (13.2), we get

$$\begin{aligned}
&\sum_{j}\mathrm{uncut}(X_j,Y_j)|E(V_j)|\\
&\qquad\le\sum_{j}\frac{4\cdot\mathrm{uncut}(S_{\mathsf{opt}}\cap V_j,T_{\mathsf{opt}}\cap V_j)|E(V_j)|}{\lambda}\le\frac{4\cdot\mathrm{uncut}(S_{\mathsf{opt}},T_{\mathsf{opt}})}{\lambda}\cdot|E|\,.
\end{aligned}$$

□

13.6 The bisection algorithm – proof of Theorem 13.1.3

First, we run the preprocessing step described in Section 13.3.1. Then we use the algorithm from Lemma 13.3.1 with $\lambda=\epsilon_{\mathsf{opt}}^{2/3}$ and $\delta=\epsilon_{\mathsf{opt}}$ to find a partition of V into sets $U_1,\dots,U_p,V_1,\dots,V_q$. We apply Corollary 13.4.4 with $\Delta=\sqrt{\epsilon_{\mathsf{opt}}}$ to all sets U_i, and obtain a list $\mathcal{U}_i$ of candidate cuts for each set U_i. Then we apply Lemma 13.4.2 and obtain a list $\mathcal{V}_j$ of candidate cuts for each set V_j. Finally, we find the optimal combination of candidate cuts using Lemma 13.5.1. Denote it by (X, Y). By Lemma 13.5.2, we get that $\mathrm{uncut}(X, Y)$ is at most

$$O\left(\sqrt{\lambda}\log(1/\delta) + \sqrt{\epsilon_{\mathsf{opt}}} + \frac{\epsilon_{\mathsf{opt}}}{\lambda} + \frac{\epsilon_{\mathsf{opt}}}{\Delta}\right) \leq O(\sqrt[3]{\epsilon_{\mathsf{opt}}}\log(1/\epsilon_{\mathsf{opt}})),$$

and

$$\left|\frac{|X|}{|V|} - \frac{1}{2}\right| \leq \max(\Delta, \delta) = O(\sqrt{\epsilon_{\mathsf{opt}}}).$$

By moving at most $O(\sqrt{\epsilon_{\mathsf{opt}}})|V|$ vertices of the smallest degree from the larger size of the cut to smaller part of the cut, we obtain a balanced cut. By doing so, we increase the number of uncut edges by at most $O(\sqrt{\epsilon_{\mathsf{opt}}}|E|)$. The obtained bisection cut cuts a $1 - O(\sqrt[3]{\epsilon_{\mathsf{opt}}}\log(1/\epsilon_{\mathsf{opt}}))$ fraction of all edges.

It is easy to see that a slight modification of the algorithm leads to the following extension of Theorem 13.1.3.

Theorem 13.6.1. *There is a randomized polynomial time algorithm that given an edge-weighted graph G with a β-biased cut of value $(1 - \epsilon)$ finds a β-biased cut of value $(1 - O(\sqrt[3]{\epsilon}\log(1/\epsilon) + \sqrt{\epsilon}/(1 - \beta)))$.*

Proof. We use the algorithm above, while changing DP algorithm used by Lemma 13.5.1 to find the best combination with bias $\beta \pm t$ (where $t = O(\sqrt{\epsilon})$). We modify the proof of Lemma 13.5.2 to show that there exists a $\beta \pm t$ cut of value $1 - O(\sqrt{\epsilon})$. As previously, we first find sets $X = X^V$ and $Y = Y^V$ with $|X^V| \geq |Y^V|$. Now, however, if $|X| - |Y| > \beta|V|$ then we add X_i' to X and Y_i' and Y; otherwise, we add X_i' to Y and Y_i' and X. We argue again that if at some point the difference $\big||X| - |Y|\big|$ becomes less than $O(\sqrt{\epsilon}|V|)$, then after that $\big||X| - |Y|\big| = O(\sqrt{\epsilon}|V|)$, and therefore, we find a cut with bias $\beta + O(\sqrt{\epsilon})$. Otherwise, there are two possible cases: either we always have $|X| - |Y| > \beta|V|$, and then we always add X_i' to X and Y_i' to Y, or we always have $|X| - |Y| > \beta|V|$, and then we always add X_i' to Y and Y_i' to X. Note, however, that in both cases $\big||X| - |Y| - \beta|V|\big|$ decreases by $|Y_i'| - |X_i'| \geq \big||S_{\mathsf{opt}} \cap U_i| - |T_{\mathsf{opt}} \cap U_i|\big| - 2\Delta|U_i|$ after each iteration. Thus after all iterations, the value of $\big||X| - |Y| - \beta|V|\big|$ decreases by at least

$$\sum_{i=1}^{p}\Big||S_{\mathsf{opt}} \cap U_i| - |T_{\mathsf{opt}} \cap U_i|\Big| - 2\Delta|U_i| \geq \Big||S_{\mathsf{opt}} \cap \bigcup_i U_i| - |T_{\mathsf{opt}} \cap \bigcup_i U_i|\Big| - 2\Delta|\bigcup_i U_i|.$$

Taking into the account that $|X^V| - |Y^V| = |S_{\mathsf{opt}} \cap \bigcup_i V_i| - |T_{\mathsf{opt}} \cap \bigcup_i V_i|$, we get the following bound for the bias of the final combination cut (X, Y),

$$
\begin{aligned}
&\Big||X| - |Y| - \beta|V|\Big| \\
&\leq \Big|\big|S_{\mathsf{opt}} \cap \bigcup_i V_i\big| - \big|T_{\mathsf{opt}} \cap \bigcup_i V_i\big| - \beta|V|\Big| - \Big|\big|S_{\mathsf{opt}} \cap \bigcup_i U_i\big| + \big|T_{\mathsf{opt}} \cap \bigcup_i U_i\big|\Big| + 2\Delta\big|\bigcup_i U_i\big| \\
&\qquad\qquad\qquad\qquad\qquad\qquad \leq \Big||S_{\mathsf{opt}}| - |T_{\mathsf{opt}}| - \beta|V|\Big| + 2\Delta|V| = 2\Delta|V|.
\end{aligned}
$$

We get the exact β-biased cut by moving at most $O(\sqrt{\epsilon})|V|$ vertices of the smallest degree from the larger size of the cut to smaller part of the cut. By doing so, we lose at most $O(\sqrt{\epsilon})|E|/(1-\beta)$ cut edges. Therefore the theorem follows. $\square$

Chapter 14

A robust isomorphism algorithm for trees

14.1 Introduction

The *graph isomorphism problem* (GISO) is arguably one of the most famous computational problems on graphs: given two graphs G and H, we have to decide whether they are *isomorphic*, i.e. there exists a bijection $\pi : V(G) \to V(H)$ such that there is an edge $(u, v) \in E(G)$ if and only if there is an edge $(\pi(u), \pi(v)) \in E(H)$. Here we use $V(G)$ to denote the set of vertices in G, and use $E(G)$ to denote the set of edges in G. GISO is one of the rare, intriguing problems in NP that is neither known to be polynomial-time tractable nor NP-Complete. Resolving the complexity of GISO is one of major problems in graph theory and it is still open despite many decades of effort.

Graph isomorphism algorithms are also very useful in practice to test for isomorphism between any structures that can be encoded as graphs. A few examples of its applications include: image analysis [82], isomorphisms of molecule (for chemistry) [84, 85], and data mining [197]. In many settings, even when the two graphs are not completely isomorphic, we are still interested in measuring how similar the two graphs are. The following optimization problem which captures the similarity between the two given graphs was studied in literature (e.g. [17, 25]).

Definition 14.1.1 (MAXGISO, rephrase of Definition 2.1.6)**.** *For two graphs G and H with the same number of vertices and a bijection $\pi : V(G) \to V(H)$, let* $\mathsf{val}(G, H, \pi) = |\{(u, v) \in E(G) \mid (\pi(u), \pi(v)) \in E(H)\}|$. *Let* $\mathsf{opt}(G, H) = \max_\pi \mathsf{val}(G, H, \pi)$. *In the* maximum graph isomorphism problem *(*MAXGISO*), the objective is to compute* $\mathsf{opt}(G, H)$

for given two graphs G and H.

Observe that GIso is a special case of the above optimization variant since $\mathsf{opt}(G,H) = \max\{|E(G)|, |E(H)|\}$ if and only if G and H are isomorphic, and any π such that $\mathsf{val}(G,H,\pi) = \max\{|E(G)|, |E(H)|\}$ is an isomorphism.

In most applications of GIso, we are interested in graphs which are *almost* isomorphic to each other. We say that G and H are $(1-\epsilon)$-*isomorphic* when $\mathsf{val}(G,H,\pi) = (1-\epsilon)\max\{E(G), E(H)\}$. In fact, even graphs which are isomorphic to each other may get perturbed slightly in practice, due to some small noise. This motivates us to introduce the following definition of *robust graph isomorphism algorithm*, which certifies when two graphs are $(1-\epsilon)$-isomorphic when ϵ is very small.

Definition 14.1.2 (RobustGIso, rephrase of Definition 2.2.7)**.** *Given two graphs G and H on n vertices, we say $\mathcal{A}$ is a* robust graph isomorphism algorithm *for G and H if there exists a function $f : [0,1] \to [0,1]$ satisfying $\lim_{\epsilon \to 0^+} f(\epsilon) = 0$, such that $\mathcal{A}$ outputs a bijection $\pi : V(G) \to V(H)$ with* $\mathsf{val}(G,H,\pi) \geq (1 - f(\epsilon))\max\{|E(G)|, |E(H)|\}$ *whenever* $\mathsf{opt}(G,H) = (1-\epsilon)\max\{|E(G)|, |E(H)|\}$*, for any $\epsilon \geq 0$.*

A robust graph isomorphism algorithm *is* a graph isomorphism algorithm (when ϵ is so small that $f(\epsilon) < \frac{1}{\max\{|E(G)|,|E(H)|\}}$). Therefore, we currently do not expect an efficient robust graph isomorphism algorithm given that a polynomial-time algorithm for GIso is not yet known. However, GIso is known to be polynomial-time tractable for many special cases such as trees [134], planar graphs [122], graphs of bounded-degree [167], and graphs of bounded tree-width [46]. This leads to the following natural question:

Question 14.1.3. *Can polynomial-time algorithms for graph isomorphism (on restricted families of instances) be made robust?*

14.1.1 Our contributionss and overview of the proofs

In this chapter, we present a robust isomorphism algorithm for trees. The well known canonicalization approach for trees seems quite sensitive to the ϵ-fraction of "noisy edges". Our algorithm is inspired by a property testing algorithm by Newman and Sohler [178] which implies a PTAS for MaxGIso on bounded-degree trees. In Section 14.2, we first show the following much weaker statement compared to Newman and Sohler's theorem, but with a simpler proof. [1]

[1]The simple proof can be modified to give robust isomorphism algorithm for bounded-degree planar graphs and graphs with bounded-width as well.

Theorem 14.1.4. *Given two n-vertex B-degree bounded forests G and H, suppose that* $\mathsf{opt}(G,H) \geq (1-\epsilon)n$. *There is a polynomial-time algorithm to find a bijection $\sigma : V(G) \to V(H)$ such that* $\mathsf{val}(G,H,\sigma) \geq (1-20\sqrt{B\epsilon})n$.

Then we reduce the general trees to bounded-degree trees, and prove that

Theorem 14.1.5. *Given two n-vertex trees G and H such that* $\mathsf{opt}(G,H) \geq (1-\epsilon)n$, *there is a polynomial-time algorithm to find a bijection $\sigma : V(G) \to V(H)$ where* $\mathsf{val}(G,H,\sigma) \geq (1-200\epsilon^{1/4})n$.

To prove Theorem 14.1.5, by removing a tiny fraction of edges, we decompose the two input trees into two collections of trees where each tree has at most one high-degree vertex. We match the high-degree vertices using the maximum weight bipartite graph matching algorithm with carefully designed weights. Then we use the algorithm in Theorem 14.1.4 to match up the low-degree vertices.

14.2 The algorithm

In this section, we prove Theorem 14.1.5, which is restated as follows.

Theorem 14.1.5 (restated). *Given two n-vertex trees G and H such that* $\mathsf{opt}(G,H) \geq (1-\epsilon)n$, *there is a polynomial-time algorithm to find a bijection $\sigma : V(G) \to V(H)$ where* $\mathsf{val}(G,H,\sigma) \geq (1-200\epsilon^{1/4})n$.

We prove Theorem 14.1.5 by first proving Theorem 14.1.4 which says that when G and H are bounded-degree forests, there is a robust isomorphism algorithm for G and H. Then, we reduce the general trees to bounded-degree trees.

Theorem 14.1.4 (restated). *Given two n-vertex B-degree bounded forests G and H, suppose that* $\mathsf{opt}(G,H) \geq (1-\epsilon)n$. *There is a polynomial-time algorithm to find a bijection $\sigma : V(G) \to V(H)$ such that* $\mathsf{val}(G,H,\sigma) \geq (1-20\sqrt{B\epsilon})n$.

Proof sketch. Let $k = \lceil 1/\sqrt{B\epsilon} \rceil$. There is a simple way to remove at most n/k edges from G to obtain a forest G', such that each tree in G' has at most kB vertices. We do the same decomposition for H to obtain a forest H'. Since we removed at most n/k edges from both G and H, we have

$$\mathsf{opt}(G',H') \geq \mathsf{opt}(G,H) - 2n/k \geq (1-\epsilon-2/k)n.$$

Now the algorithm lets $G'' = G'$, $H'' = H'$. The algorithm chooses a tree T_G from G'' and a tree T_H from H'' so that T_G is isomorphic to T_H, lets σ map the vertices in T_G to the vertices in T_H according to the isomorphism, and removes T_G from G'', T_H from H''. The algorithm iterates this process until no such pair of trees can be found in G'' and H''. Finally the algorithm extends σ to a bijection between vertex sets of G and H.

It is easy to see that, when the algorithm terminates, the total number of trees in G'' is at most $4(|E(G')| + |E(H')| - 2\mathsf{opt}(G', H')) \leq 8\epsilon n$ (because $|E(G')| \leq (1 - 2/k)n$ and $|E(H')| \leq (1 - 2/k)n$). Since each tree has at most kB edges, σ loses at most $kB \cdot 8\epsilon n$ edges. Therefore,

$$\mathsf{val}(G, H, \sigma) \geq \mathsf{val}(G', H', \sigma) \geq \mathsf{opt}(G', H') - 8kB\epsilon n \geq (1 - 2/k - 9kB\epsilon)n \geq (1 - 20\sqrt{B\epsilon})n.$$

□

Now we introduce the following definition.

Definition 14.2.1. *We call a tree T a B-*tree *if there is one vertex with degree at least B, while other vertices have degree less than B. The vertex with highest degree is referred to as the* center *of T. Let $\mathrm{cdeg}(T)$ be the degree of the center.*

We will use Theorem 14.1.4 to prove the following Lemma 14.2.2, which says that there is an robust isomorphism algorithm for forests of B-trees and degree bounded trees.

Lemma 14.2.2. *Let G and H be two n-vertex forests of B-trees and $(B - 1)$-degree bounded trees. Given that $\mathsf{opt}(G, H) \geq |E(G)| - \epsilon n$, there is a polynomial-time algorithm to find a bijection $\sigma : V(G) \to V(H)$ such that $\mathsf{val}(G, H, \sigma) \geq \mathsf{opt}(G, H) - 100B\sqrt{\epsilon}n$.*

We defer the proof of Lemma 14.2.2 to the next subsection. Now we prove Theorem 14.1.5 using Lemma 14.2.2.

Proof of Theorem 14.1.5 from Lemma 14.2.2. For any integer parameter $B \geq 2$, one can remove at most $2n/B$ edges from G to get a forest of B-trees and $(B - 1)$-degree bounded trees, namely G'. (To see this, simply root G using an arbitrary vertex, and for each vertex whose degree is no less than B, remove the edge to its parent.) We do the similar decomposition for H to get H'. Since we removed at most n/B edges from both G and H, we have $\mathsf{opt}(G', H') \geq \mathsf{opt}(G, H) - 4n/B \geq (1 - \epsilon - 4/B)n$. Also observe that $|E(G')| - \mathsf{opt}(G', H') \leq |E(G)| - \mathsf{opt}(G, H) \leq \epsilon n$.

Now we apply the algorithm in Lemma 14.2.2 to G' and H' to get a bijection σ. We have

$$\mathsf{val}(G, H, \sigma) \geq \mathsf{val}(G', H', \sigma) \geq \mathsf{opt}(G', H') - 100B\sqrt{\epsilon}n \geq (1 - \epsilon - 4/B - 100B\sqrt{\epsilon})n.$$

If we take $B = \lceil \epsilon^{-1/4} \rceil$, the algorithm finds a bijection σ such that $\mathsf{val}(G, H, \sigma) \geq (1 - 200\epsilon^{1/4})n$.

$\square$

14.2.1 Robust isomorphism algorithm for B-trees

In this subsection, we prove Lemma 14.2.2. Let G consist of $G_1, G_2, \ldots, G_p$ and $\tilde{G}$ where $G_1, G_2, \ldots, G_p$ are B-trees and $\tilde{G}$ is a forest of $(B-1)$-degree bounded trees; let H consist of $H_1, H_2, \ldots, H_q$ and $\tilde{H}$ where $H_1, H_2, \ldots, H_q$ are B-trees and $\tilde{H}$ is a forest of $(B-1)$-degree bounded trees.

For any two B-trees G_i and H_j, let $E_c(G_i)$ and $E_c(H_j)$ be the set of edges incident to the centers of G_i and H_j. Now let $C(G_i, H_j)$ be the minimum number of edges in $E_c(G_i) \cup E_c(H_j)$ one has to remove from G_i and H_j, so that the connected components where the two centers are in are isomorphic and the isomorphism maps the center of G_i to the center of H_j. Also, let $C_e(G_i, H_j)$ be the set of edges removed from $E_c(G_i) \cup E_c(H_j)$. It is easy to see that both $C(\cdot, \cdot)$ and $C_e(\cdot, \cdot)$ can be computed in polynomial-time.

The algorithm. The first step of the algorithm finds a (partial) matching between $\{G_1, G_2, \ldots, G_p\}$ and $\{H_1, H_2, \ldots, H_q\}$. We use two mappings

$$\tau : \{G_1, G_2, \ldots, G_p\} \to \{H_1, H_2, \ldots, H_q, \perp\}, \text{ and}$$
$$\tau' : \{H_1, H_2, \ldots, H_q\} \to \{G_1, G_2, \ldots, G_p, \perp\}$$

to denote the (partial) matching. It is satisfied that $\tau(G_i) = H_j$ iff $\tau'(H_j) = G_i$. The algorithm uses the (polynomial-time) maximum weight bipartite graph matching algorithm to find τ and τ' so that the following cost is minimized.

$$\begin{aligned}\mathrm{cost}(\tau, \tau') = &\sum_{G_i : \tau(G_i) = \perp} (\mathrm{cdeg}(G_i) - B + 1) \\ &+ \sum_{H_j : \tau'(H_j) = \perp} (\mathrm{cdeg}(H_j) - B + 1) + \sum_{G_i : \tau(G_i) \neq \perp} C(G_i, \tau(G_i)).\end{aligned}$$

For each G_i, let $E'_c(G_i)$ be an arbitrary subset of cardinality $(\mathrm{cdeg}(G_i) - B + 1)$ of $E_c(G_i)$. Define $E'_c(H_j)$ similarly for each H_j. Let

$$E_0 = \bigcup_{G_i : \tau(G_i) = \perp} E'_c(G_i) \bigcup \bigcup_{H_j : \tau'(H_j) = \perp} E'_c(H_j) \bigcup \bigcup_{G_i : \tau(G_i) \neq \perp} C_e(G_i, \tau(G_i)),$$

we have $|E_0| = \mathrm{cost}(\tau, \tau')$.

In the second step of the algorithm, we remove the edges in E_0 from G and H, call the remaining graphs G' and H' respectively. Observe that $\mathsf{opt}(G', H') \geq \mathsf{opt}(G, H) - |E_0| = \mathsf{opt}(G, H) - \mathrm{cost}(\tau, \tau')$. Both G' and H' can be divided into two parts: $G^{(a)}, H^{(a)}$ and $G^{(b)}, H^{(b)}$. $G^{(a)}$ consists of the connected components where the centers of G_i's are in, where $\tau(G_i) \neq \perp$. Similarly, $H^{(a)}$ consists of the connected components where the centers of H_j's are in, where $\tau'(H_j) \neq \perp$. $G^{(b)}$ and $H^{(b)}$ are the remaining parts of G' and H'. Observe that $G^{(a)}$ is isomorphic to $H^{(a)}$, while $G^{(b)}$ and $H^{(b)}$ are $(B-1)$-degree bounded.

Finally, we use the algorithm in Theorem 14.1.4 to find an approximation to $\mathsf{opt}(G^{(b)}, H^{(b)})$. Suppose that each of $G^{(b)}$, $H^{(b)}$ has $n^{(b)}$ vertices. The algorithm finds a bijection $\sigma^{(b)}$ such that

$$\begin{aligned}\mathsf{val}(G^{(b)}, H^{(b)}, \sigma^{(b)}) &\geq \left(1 - 20\sqrt{\frac{B(n^{(b)} - \mathsf{opt}(G^{(b)}, H^{(b)}))}{n^{(b)}}}\right) n^{(b)} \\ &\geq \mathsf{opt}(G^{(b)}, H^{(b)}) - 20\sqrt{B(n^{(b)} - \mathsf{opt}(G^{(b)}, H^{(b)}))n^{(b)}}. \quad (14.1)\end{aligned}$$

Observe that since $\mathsf{opt}(G^{(b)}, H^{(b)}) + |E_0| + |E(G^{(a)})| \geq \mathsf{opt}(G, H) \geq (1-\epsilon)n$, we have

$$\begin{aligned}n^{(b)} - \mathsf{opt}(G^{(b)}, H^{(b)}) \leq n^{(b)} - n + |E(G^{(a)})| + |E_0| + \epsilon n &\leq \epsilon n + |E_0| \\ &= \epsilon n + \mathrm{cost}(\tau, \tau'), \quad (14.2)\end{aligned}$$

where the last inequality is because $|E(G^{(a)})|$ is less than the number of vertices in $G^{(a)}$, which is $n - n^{(b)}$. Now we combine (14.1) and (14.2), getting

$$\begin{aligned}\mathsf{val}(G^{(b)}, H^{(b)}, \sigma^{(b)}) &\geq \mathsf{opt}(G^{(b)}, H^{(b)}) - 20\sqrt{B(\epsilon n + \mathrm{cost}(\tau, \tau'))n^{(b)}} \\ &\geq \mathsf{opt}(G^{(b)}, H^{(b)}) - 20\sqrt{B(\epsilon n + \mathrm{cost}(\tau, \tau'))n}. \quad (14.3)\end{aligned}$$

Let $\sigma^{(a)}$ be the isomorphism between $G^{(a)}$ and $H^{(a)}$. The algorithm lets $\sigma = \sigma^{(a)} \cup \sigma^{(b)}$ be the final solution.

Analysis. The analysis of the algorithm uses the following lemmas.

Lemma 14.2.3. $\mathrm{cost}(\tau, \tau') \leq 2(B+1)\epsilon n \leq 4B\epsilon n$.

Lemma 14.2.4. *Let $G = G_0 \cup G_1$ be a union of two graphs with disjoint vertex sets. Similarly, let $H = H_0 \cup H_1$. Suppose that G_0 is isomorphic to H_0 and $|V(G)| = |V(H)|$, we have* $\mathsf{opt}(G, H) = \mathsf{opt}(G_1, H_1) + |E(G_0)|$.

We prove both Lemma 14.2.3 and Lemma 14.2.4 in the next subsection. Lemma 14.2.4 is very intuitive: if G has a connected component G_0 which is isomorphic to a connected component H_0 in H, there must be an optimal solution for G and H which maps G_0 to H_0.

Now we proceed to analyze $\mathsf{val}(G, H, \sigma)$ using these lemmas.

$$\begin{aligned}
\mathsf{val}(G, H, \sigma) &\geq \mathsf{val}(G^{(a)}, H^{(a)}, \sigma^{(a)}) + \mathsf{val}(G^{(b)}, H^{(b)}, \sigma^{(b)}) \\
&\geq |E(G^{(a)})| + \mathsf{opt}(G^{(b)}, H^{(b)}) \\
&\qquad - 20\sqrt{B(\epsilon n + \mathrm{cost}(\tau, \tau'))n} && \text{(by (14.3))} \\
&\geq |E(G^{(a)})| + \mathsf{opt}(G^{(b)}, H^{(b)}) - 20B\sqrt{5\epsilon}n && \text{(by Lemma 14.2.3)} \\
&= \mathsf{opt}(G', H') - 20B\sqrt{5\epsilon}n && \text{(Lemma 14.2.4)} \\
&\geq \mathsf{opt}(G, H) - 4B\epsilon n - 20B\sqrt{5\epsilon}n && \text{(by Lemma 14.2.3 again)} \\
&\geq \mathsf{opt}(G, H) - 100B\sqrt{\epsilon}n. && \square
\end{aligned}$$

14.2.2 Proofs of Lemma 14.2.3 and Lemma 14.2.4

Proof of Lemma 14.2.3. Let $\sigma^* : V(G) \to V(H)$ be a bijection such that $\mathsf{val}(G, H, \sigma^*) = (1-\epsilon)n$. We define $\tau^* : \{G_1, G_2, \ldots, G_p\} \to \{H_1, H_2, \ldots, H_q, \perp\}$ and $\tau^{*\prime} : \{H_1, H_2, \ldots, H_q\} \to \{G_1, G_2, \ldots, G_p, \perp\}$ as follows.

- $\tau^*(G_i) = H_j$ when σ^* maps the center of G_i to the center of H_j; $\tau^*(G_i) = \perp$ when the center of G_i is not mapped to the center of any H_j.

- $\tau^{*\prime}(H_j) = G_i$ when σ^* maps the center of G_i to the center of H_j; $\tau^{*\prime}(H_j) = \perp$ when none of the centers of G_i is not mapped to the center of H_j.

Now we are going to upper bound $\mathrm{cost}(\tau^*, \tau^{*\prime})$ and therefore prove the lemma.

For each G_i where $1 \leq i \leq p$, if the center of G_i is not mapped by σ^* to the center of any H_j, it must be mapped to a vertex whose degree is less than B. This means that at least $(\mathrm{cdeg}(G_i) - B + 1)$ edges incident to the center of G_i are not mapped to H. Therefore, we have

$$\sum_{G_i : \tau^*(G_i) = \perp} (\mathrm{cdeg}(G_i) - B + 1) \leq \epsilon n.$$

Since $\mathrm{cdeg}(G_i)/(\mathrm{cdeg}(G_i) - B + 1) \leq B$, we have

$$\sum_{G_i : \tau^*(G_i) = \perp} \mathrm{cdeg}(G_i) \leq \epsilon B n \tag{14.4}$$

Similarly, we have

$$\sum_{H_j : \tau^{*\prime}(H_j) = \perp} \mathrm{cdeg}(H_j) \leq \epsilon B n. \tag{14.5}$$

For each $1 \leq i \leq p, 1 \leq j \leq q$, let E_{G_i} be the set of the edges in G_i that are not mapped to H; let E_{H_j} be the set of the edges in H_j that are not mapped to G. Now consider any pair of G_i and H_j such that σ^* maps the center of G_i to the center of H_j. Recall that $C(G_i, H_j)$ is the minimum number of edges in $E_c(G_i) \cup E_c(H_j)$ one has to remove from G_i and H_j, so that the connected components corresponding to the two centers are isomorphic. Also recall that $E_c(G_i)$ and $E_c(H_j)$ are the sets of edges incident to the centers of G_i and H_j respectively.

For each edge $e \in E_{G_i}$, we remove the corresponding edge in $E_c(G_i)$, i.e. the unique edge on the unique path connecting e to the center of G_i. Similarly, for each edge $e \in E_{H_j}$, we remove the corresponding edge in $E_c(H_j)$, i.e. the unique edge on the unique path connecting e to the center of H_j. In total, at most $|E_{G_i}| + |E_{H_j}|$ edges are removed.

We observe that, after removing the edges, the connected components corresponding to the centers of G_i and H_j are isomorphic and the isomorphism maps the center of G_i to the center of H_j : σ^* defines such an isomorphism.

To summarize, we have proved that $C(G_i, H_j) \leq |E_{G_i}| + |E_{H_j}|$. Therefore,

$$\sum_{G_i : \tau^*(G_i) \neq \perp} C(G_i, \tau(G_i)) \leq \sum_{G_i : \tau^*(G_i) \neq \perp} (|E_{G_i}| + |E_{\tau^*(G_i)}|) \leq \sum_{i=1}^{p} |E_{G_i}| + \sum_{j=1}^{q} |E_{H_j}| \leq 2\epsilon n. \tag{14.6}$$

Now, summing up (14.4), (14.5), and (14.6), we get

$$\begin{aligned} &\mathrm{cost}(\tau, \tau') \leq \mathrm{cost}(\tau^*, \tau^{*\prime}) \\ = &\sum_{G_i : \tau^*(G_i) = \perp} \mathrm{cdeg}(G_i) + \sum_{H_j : \tau^{*\prime}(H_j) = \perp} \mathrm{cdeg}(H_j) + \sum_{G_i : \tau^*(G_i) \neq \perp} C(G_i, \tau(G_i)) \leq 2(B+1)\epsilon n. \end{aligned}$$

□

Proof of Lemma 14.2.4. It is straightforward to see that $\mathsf{opt}(G, H) \geq \mathsf{opt}(G_1, H_1) + |E(G_0)|$. Now we prove that $\mathsf{opt}(G, H) \leq \mathsf{opt}(G_1, H_1) + |E(G_0)|$.

Given $\sigma : V(G) \to V(H)$ such that $\mathsf{val}(G, H, \sigma) = \mathsf{opt}(G, H)$, we define the bijection $\sigma' : V(G_1) \to V(H_1)$ as follows and prove that $\mathsf{opt}(G, H) \leq \mathsf{val}(G_1, H_1, \sigma') + |E(G_0)|$.

Let the bijection $\tau : V(G_0) \to V(H_0)$ be the isomorphism between G_0 and H_0. For each $v \in V(G_1)$, use the following procedure to decide $\sigma'(v)$: if $\sigma(v) \in V(H_1)$, then return $\sigma(v)$; if $\sigma(v) \in V(H_0)$, repeat this procedure with $v \leftarrow \tau^{-1}(\sigma(v))$.

One can verify that the procedure above always terminates for all $v \in V(G_1)$, and that σ' is indeed a bijection. It remains to show that $\mathsf{val}(G_1, H_1, \sigma') \geq \mathsf{opt}(G, H) - |E(G_0)|$.

For each edge $(u, v) \in E(G_1)$ such that $(\sigma(u), \sigma(v)) \in E(H)$, let us write down $u_1 = \sigma(u), u_2, u_3, \ldots, u_p = \sigma'(u)$ and $v_1 = \sigma(v), v_2, v_3, \ldots, v_q = \sigma'(v)$ to be the sequence of vertices in $V(H)$ visited by the procedure above when the input is u and v respectively. We can assume w.l.o.g. that $p \leq q$.

Suppose that $(\sigma'(u), \sigma'(v)) \notin E(H_1)$, we know that $u_p = \sigma'(u) \in V(H_1)$ and $v_p \in V(H_0)$, and therefore $(u_p, v_p) \notin E(H)$. Let i be the smallest index such that $(u_i, v_i) \notin E(H)$. We have that $i \geq 2$. Therefore $(u_{i-1}, v_{i-1}) \in E(H_0)$. Since τ is the isomorphism between G_0 and H_0, we have that $(\tau^{-1}(u_{i-1}), \tau^{-1}(v_{i-1})) \in E(G_0)$. Also note that $(\sigma(\tau^{-1}(u_{i-1})), \sigma(\tau^{-1}(v_{i-1}))) = (u_i, v_i) \notin E(H)$.

To summarize, for each $(u, v) \in E(G_1)$ such that $(\sigma(u), \sigma(v)) \in E(H)$ and $(\sigma'(u), \sigma'(v)) \notin E(H_1)$, we have set up a mapping $f((u, v)) = (u_f, v_f) = (\tau^{-1}(u_{i-1}), \tau^{-1}(v_{i-1}))$ with the property that $(u_f, v_f) \in E(G_0)$ and $(\sigma(u_f), \sigma(v_f)) \notin E(H)$. By the fact that the procedure to define $\sigma'(v)$ is reversible at any point, one can verify that f is injective. Therefore,

$$
\begin{aligned}
&|\{(u, v) \in E(G_1) : (\sigma(u), \sigma(v)) \in E(H), (\sigma'(u), \sigma'(v)) \notin E(H_1))\}| \\
&\qquad\qquad\qquad \leq |\{(u, v) \in E(G_0) : (\sigma(u), \sigma(v)) \notin E(H)\}|.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&\mathsf{val}(G_1, H_1, \sigma') = |E(G_1)| - |\{(u, v) \in E(G_1) : (\sigma'(u), \sigma'(v)) \notin E(H_1))\}| \\
\geq &|E(G_1)| - |\{(u, v) \in E(G_1) : (\sigma'(u), \sigma'(v)) \notin E(H_1)), (\sigma(u), \sigma(v)) \in E(H)\}| \\
&\quad - |\{(u, v) \in E(G_1) : (\sigma(u), \sigma(v)) \notin E(H)\}| \\
\geq &|E(G_1)| - |\{(u, v) \in E(G_0) : (\sigma(u), \sigma(v)) \notin E(H)\}| - \\
&\quad |\{(u, v) \in E(G_1) : (\sigma(u), \sigma(v)) \notin E(H)\}| \\
= &|E(G_1)| - |E(G_0)| + |\{(u, v) \in E(G_0) : (\sigma(u), \sigma(v)) \in E(H)\}| \\
&\quad - |\{(u, v) \in E(G_1) : (\sigma(u), \sigma(v)) \notin E(H)\}|
\end{aligned}
$$

$$
\begin{aligned}
&= \mathsf{val}(G, H, \sigma) - |E(G_0)| \\
&= \mathsf{opt}(G, H) - |E(G_0)|.
\end{aligned}
$$

$\square$

Part IV

Other approximation and hardness of approximation results

Chapter 15

Certifying the $2 \to 4$ norm of random linear operators

15.1 Introduction

For a function $f\colon \Omega \to \mathbb{R}$ on a (finite) probability space Ω, the *p-norm* is defined as $\|f\|_p = (\mathbf{E}_\Omega f^p)^{1/p}$.[1] The *$p \to q$ norm* $\|A\|_{p\to q}$ of a linear operator A between vector spaces of such functions is the smallest number $c \geq 0$ such that $\|Af\|_q \leq c\,\|f\|_p$ for all functions f in the domain of A. We also define the *$p \to q$ norm* of a *subspace* V to be the maximum of $\|f\|_q / \|f\|_p$ for $f \in V$; note that for $p = 2$ this is the same as the norm of the projector operator into V.

In this chapter, we are interested in the case $p < q$ and we will call such $p \to q$ norms *hypercontractive*.[2] Roughly speaking, for $p < q$, a function f with large $\|f\|_q$ compared to $\|f\|_p$ can be thought of as "spiky" or somewhat sparse (i.e., much of the mass concentrated in small portion of the entries). Hence finding a function f in a linear subspace V maximizing $\|f\|_q / \|f\|_2$ for some $q > 2$ can be thought of as a geometric analogue of the problem finding the shortest word in a linear code. This problem is equivalent to computing the $2 \to q$ norm of the projector P into V (since $\|Pf\|_2 \leq \|f\|_2$). Also when A is a normalized adjacency matrix of a graph (or more generally a Markov operator), upper

[1]We follow the convention to use *expectation* norms for *functions* (on probability spaces) and *counting* norms, denoted as $\|v\|_p = (\sum_{i=1}^n |v_i|^p)^{1/p}$, for *vectors* $v \in \mathbb{R}^m$. All normed spaces here will be finite dimensional. We distinguish between expectation and counting norms to avoid recurrent normalization factors.

[2]We use this name because a bound of the form $\|A\|_{p\to q} \leq 1$ for $p < q$ is often called a *hypercontractive inequality*.

bounds on the $p \to q$ norm are known as *mixed-norm*, *Nash* or *hypercontractive inequalities* and can be used to show rapid mixing of the corresponding random walk (e.g., see the surveys [104, 201]). Such bounds also have many applications to theoretical computer science, which are described in the survey [43].

However, very little is known about the complexity of computing these norms. This is in contrast to the case of $p \to q$ norms for $p \geq q$, where much more is known both in terms of algorithms and lower bounds, see [214, 146, 42].

We study a natural semidefinite programming (SDP) relaxation for computing the $2 \to 4$ norm of a given linear operator which we call TensorSDP.[3] While TensorSDP is very unlikely to provide a poly-time constant-factor approximation for the $2 \to 4$ norm in general (as shown in [34]), we do show that it provides such approximation on random linear operators, as we describe below.

We show that TensorSDP certifies a constant upper bound on the ratio $\|A\|_{2\to 4}/\left\|A\right\|_{2\to 2}$ where $A : \mathbb{R}^n \to \mathbb{R}^m$ is a *random linear operator* (e.g., obtained by a matrix with entries chosen as i.i.d Bernoulli variables) and $m \geq \Omega(n^2)$. In contrast, if $m = o(n^2)$ then this ratio is $\omega(1)$, and hence this result is almost tight in the sense of obtaining "good approximation" in the sense mentioned above. We find this interesting, since random matrices seem like natural instances; indeed for superficially similar problems such shortest codeword, shortest lattice vector (or even the $1 \to 2$ norm), it seems hard to efficiently certify bounds on random operators.

15.2 The TensorSDP algorithm

Observe that when the linear operator A is given in the form of $A = \sum_{j=1}^{m} e_j a_j^T$, for every x such that $\left\|x\right\|_2 = 1$, we have

$$\|Ax\|_4^4 = \frac{1}{m}\sum_{j=1}^{m}(a_j^T x)^4 = \frac{1}{m}\sum_{j=1}^{m}\sum_{i_1,i_2,i_3,i_4\in[n]} \operatorname{tr}(a_i a_i^T \otimes a_i a_i^T)(xx^T \otimes xx^T).$$

Instead of maximizing the above quantity over all $xx^T \otimes xx^T$, we maximize over all the 4-th moment matrices. This gives the following natural semidefinite program for estimating the $2 \to 4$ norm of A.

[3]We use the name TensorSDP for this program since it will be a canonical relaxation of the polynomial program $\max_{\|x\|_2=1} \langle T, x^{\otimes 4}\rangle$ where T is the 4-tensor such that $\langle T, x^{\otimes 4}\rangle = \|Ax\|_4^4$. Indeed, it is identical to the degree-2 Lasserre–Parrilo SDP relaxation.

Algorithm 2 The TensorSDP algorithm for $2 \to 4$ norms.

Input: A linear operator $A = \sum_{j=1}^m e_j a_j^T$.
Output: An estimated value for $\|A\|_{2\to4}^4$.
Find $X = (X_{(i_1,i_2),(i_3,i_4)})_{i_1,i_2,i_3,i_4\in[n]}$ over set of $n^2 \times n^2$ real matrices to maximize

$$\operatorname{tr}\left(\frac{1}{m}\sum_{j=1}^m \sum_{i_1,i_2,i_3,i_4\in[n]} \operatorname{tr}(a_i a_i^T \otimes a_i a_i^T)\right) X$$

such that

- $X \succeq 0$,
- $\mathbf{E}_{i,j\in[n]} X_{(i,j),(i,j)} = 1$,
- $X_{(i_1,i_2),(i_3,i_4)} = X_{(\pi(i_1),\pi(i_2)),(\pi(i_3),\pi(i_4))}$ for all permutations π over $[4]$.

Output the above quantity with the maximizer X.

15.3 Certifying the hypercontractivity of random operators

In this section we show that TensorSDP provides non-trivial approximation guarantees on the $2 \to 4$ norms of random linear operators.

Let $A = \sum_{i=1}^m e_i a_i^T/\sqrt{n}$, where e_i is the vector with a 1 in the i^{th} position, and each a_i is chosen i.i.d. from a distribution $\mathcal{D}$ on $\mathbb{R}^n$. Three natural possibilities are

1. $\mathcal{D}_{\text{sign}}$: the uniform distribution over $\{-1,1\}^n$
2. $\mathcal{D}_{\text{Gaussian}}$: a vector of n independent Gaussians with mean zero and variance 1
3. $\mathcal{D}_{\text{unit}}$: a uniformly random (expectation-norm) unit vector on $\mathbb{R}^n$.

Our arguments will apply to any of these cases, or even to more general nearly-unit vectors with bounded sub-Gaussian moment (details below).

Before discussing the performance of TensorSDP, we will discuss how the $2 \to 4$-norm of A behaves as a function of n and m. We can gain intuition by considering two limits in the case of $\mathcal{D}_{\text{Gaussian}}$. If $n = 1$, then $\|A\|_{2\to4} = \|a\|_4$, for a random Gaussian vector

a. For large m, $\|a\|_4$ is likely to be close to $3^{1/4}$, which is the fourth moment of a mean-zero unit-variance Gaussian. By Dvoretzky's theorem [186], this behavior can be shown to extend to higher values of n. Indeed, there is a universal $c > 0$ such that if $n \leq c\sqrt{m}\epsilon^2$, then w.h.p. $\|A\|_{2\to 4} \leq 3^{1/4} + \epsilon$. In this case, the maximum value of $\|Ax\|_4$ looks roughly the same as the average or the minimum value, and we also have $\|Ax\|_4 \geq (3^{1/4} - \epsilon)\,\|x\|_2$ for all $x \in \mathbb{R}^n$. In the cases of $\mathcal{D}_{\text{sign}}$ and $\mathcal{D}_{\text{unit}}$, the situation is somewhat more complicated, but for large n, their behavior becomes similar to the Gaussian case.

On the other hand a simple argument (see e.g. [34]) shows that $\|A\|_{2\to 4} \geq n^{1/2}/m^{1/4}$ for *any* (not only random) $m \times n$ matrix with all $\pm 1/\sqrt{n}$ entries. A nearly identical bound applies for the case when the a_i are arbitrary unit or near-unit vectors. Thus, in the regime where $n \geq \omega(\sqrt{m})$, we always have $\|A\|_{2\to 4} \geq \omega(1)$.

The following theorem shows that $\mathsf{TensorSDP}$ achieves approximately the correct answer in both regimes.

Theorem 15.3.1. *Let $a_1, \ldots, a_m$ be drawn i.i.d. from a distribution $\mathcal{D}$ on $\mathbb{R}^n$ with $\mathcal{D} \in \{\mathcal{D}_{Gaussian}, \mathcal{D}_{sign}, \mathcal{D}_{unit}\}$, and let $A = \sum_{i=1}^m e_i a_i^T/\sqrt{n}$. Then w.h.p.* $\mathsf{TensorSDP}(A) \leq 3 + c\max(\frac{n}{\sqrt{m}}, \frac{n^2}{m})$ *for some constant $c > 0$.*

From Theorem 15.3.1 and the fact that $\|A\|_{2\to 4}^4 \leq \mathsf{TensorSDP}(A)$, we obtain:

Corollary 15.3.2. *Let A be as in Theorem 15.3.1. Then $\exists c > 0$ such that w.h.p.*

$$\|A\|_{2\to 4} \leq \begin{cases} 3^{1/4} + c\frac{n}{\sqrt{m}} & \text{if } n \leq \sqrt{m} \\ c\frac{n^{1/2}}{m^{1/4}} & \text{if } n > \sqrt{m} \end{cases} \tag{15.1}$$

Before proving Theorem 15.3.1, we introduce some more notation. This will in fact imply that Theorem 15.3.1 applies to a broader class of distributions. For a distribution $\mathcal{D}$ on $\mathbb{R}^N$, define the ψ_p norm $\|\mathcal{D}\|_{\psi_p}$ to be the smallest $C > 0$ such that

$$\max_{v \in S(\mathbb{R}^N)} \mathop{\mathbf{E}}_{a \sim \mathcal{D}} e^{\frac{|\langle v,a\rangle|^p N^{p/2}}{C^p}} \leq 2, \tag{15.2}$$

or ∞ if no finite such C exists. We depart from the normal convention by including a factor of $N^{p/2}$ in the definition, to match the scale of [2]. The ψ_2 norm (technically a seminorm) is also called the sub-Gaussian norm of the distribution. One can verify that for each of the above examples (sign, unit and Gaussian vectors), $\psi_2(\mathcal{D}) \leq O(1)$.

We also require that $\mathcal{D}$ satisfies a *boundedness* condition with constant $K \geq 1$, defined as

$$\Pr\left[\max_{i\in[m]} \|a_i\|_2 > K\max(1, (m/N)^{1/4})\right] \leq e^{-\sqrt{N}}. \tag{15.3}$$

Similarly, K can be taken to be $O(1)$ in each case that we consider.

We will require a following result of [1, 2] about the convergence of sums of i.i.d rank-one matrices.

Lemma 15.3.3 ([2]). *Let $\mathcal{D}'$ be a distribution on $\mathbb{R}^N$ such that $\mathbf{E}_{v\sim\mathcal{D}'} vv^T = I$, $\|\mathcal{D}'\|_{\psi_1} \le \psi$ and* (15.3) *holds for $\mathcal{D}'$ with constant K. Let $v_1, \ldots, v_m$ be drawn i.i.d. from $\mathcal{D}'$. Then with probability $\ge 1 - 2\exp(-c\sqrt{N})$, we have*

$$(1-\epsilon)I \le \frac{1}{m}\sum_{i=1}^{m} v_i v_i^T \le (1+\epsilon)I, \tag{15.4}$$

where $\epsilon = C(\psi + K)^2 \max(N/m, \sqrt{N/m})$ with $c, C > 0$ universal constants.

The $N \le m$ case (when the $\sqrt{N/m}$ term is applicable) was proven in Theorem 1 of [2], and the $N > m$ case (i.e. when the max is achieved by N/m) was proven in Theorem 2 of [2] (see also Theorem 3.13 of [1]).

Proof of Theorem 15.3.1. Define $A_{2,2} = \frac{1}{m}\sum_{i=1}^{m} a_i a_i^T \otimes a_i a_i^T$. For $n^2 \times n^2$ real matrices X, Y, define $\langle X, Y\rangle := \mathrm{tr} X^T Y/n^2 = \mathbf{E}_{i,j\in[n]} X_{i,j}Y_{i,j}$. Additionally define the convex set $\mathcal{X}$ to be the set of $n^2 \times n^2$ real matrices $X = (X_{(i_1,i_2),(i_3,i_4)})_{i_1,i_2,i_3,i_4\in[n]}$ with $X \succeq 0$, $\mathbf{E}_{i,j\in[n]} X_{(i,j),(i,j)} = 1$ and $X_{(i_1,i_2),(i_3,i_4)} = X_{(i_{\pi(1)},i_{\pi(2)}),(i_{\pi(3)},i_{\pi(4)})}$ for any permutation $\pi \in \mathcal{S}_4$. Finally, let $h_{\mathcal{X}}(Y) := \max_{X\in\mathcal{X}}\langle X, Y\rangle$. It is straightforward to show that

$$\mathsf{TensorSDP}(A) = h_{\mathcal{X}}(A_{2,2}) = \max_{X\in\mathcal{X}}\langle X, A_{2,2}\rangle. \tag{15.5}$$

We note that if $\mathcal{X}$ were defined without the symmetry constraint, it would simply be the convex hull of xx^T for unit vectors $x \in \mathbb{R}^{n^2}$ and $\mathsf{TensorSDP}(A)$ would simply be the largest eigenvalue of $A_{2,2}$. However, we will later see that the symmetry constraint is crucial to $\mathsf{TensorSDP}(A)$ being $O(1)$.

Our strategy will be to analyze $A_{2,2}$ by applying Lemma 15.3.3 to the vectors $v_i := \Sigma^{-1/2}(a_i \otimes a_i)$, where $\Sigma = \mathbf{E}\, a_i a_i^T \otimes a_i a_i^T$, and $^{-1/2}$ denotes the pseudo-inverse. First, observe that, just as the ψ_2 norm of the distribution over a_i is constant, a similar calculation can verify that the ψ_1 norm of the distribution over $a_i \otimes a_i$ is also constant. Next, we have to argue that $\Sigma^{-1/2}$ does not increase the norm by too much.

To do so, we compute Σ for each distribution over a_i that we have considered. Let F be the operator satisfying $F(x \otimes y) = y \otimes x$ for any $x, y \in \mathbb{R}^n$; explicitly $F = \sum_{i,j=1}^{n} e_i e_j^T \otimes e_j e_i^T$.

Define

$$\Phi := \sum_{i=1}^{n} e_i \otimes e_i \tag{15.6}$$

$$\Delta := \sum_{i=1}^{n} e_i e_i^T \otimes e_i e_i^T \tag{15.7}$$

Direct calculations (omitted) can verify that the cases of random Gaussian vectors, random unit vectors and random ± 1 vectors yield respectively

$$\Sigma_{\text{Gaussian}} = I + F + \Phi\Phi^T \tag{15.8a}$$

$$\Sigma_{\text{unit}} = \frac{n}{n+1}\Sigma_{\text{Gaussian}} \tag{15.8b}$$

$$\Sigma_{\text{sign}} = \Sigma_{\text{Gaussian}} - 2\Delta \tag{15.8c}$$

In each case, the smallest nonzero eigenvalue of Σ is $\Omega(1)$, so $v_i = \Sigma^{-1/2}(a_i \otimes a_i)$ has $\psi_1 \leq O(1)$ and satisfies the boundedness condition (15.3) with $K \leq O(1)$.

Thus, we can apply Lemma 15.3.3 (with $N = \operatorname{rank}\Sigma \leq n^2$ and $\epsilon := c\max(n/\sqrt{m}, n^2/m)$) and find that in each case w.h.p.

$$A_{2,2} = \frac{1}{m}\sum_{i=1}^{m} a_i a_i^T \otimes a_i a_i^T \preceq (1+\epsilon))\,\Sigma \preceq (1+\epsilon)\left(I + F + \Phi\Phi^T\right) \tag{15.9}$$

Since $h_{\mathcal{X}}(Y) \geq 0$ whenever $Y \succeq 0$, we have $h_{\mathcal{X}}(A_{2,2}) \leq (1+\epsilon)h_{\mathcal{X}}(\Sigma)$. Additionally, $h_{\mathcal{X}}(I + F + \Phi\Phi^T) \leq h_{\mathcal{X}}(I) + h_{\mathcal{X}}(F) + h_{\mathcal{X}}\Phi\Phi^T)$, so we can bound each of three terms separately. Observe that I and F each have largest eigenvalue equal to 1, and so $h_{\mathcal{X}}(I) \leq 1$ and $h_{\mathcal{X}}(F) \leq 1$. (In fact, these are both equalities.)

However, the single nonzero eigenvalue of $\Phi\Phi^T$ is equal to n. Here we will need to use the symmetry constraint on $\mathcal{X}$. Let X^Γ be the matrix with entries $X^\Gamma_{(i_1,i_2),(i_3,i_4)} := X_{(i_1,i_4),(i_3,i_2)}$. If $X \in \mathcal{X}$ then $X = X^\Gamma$. Additionally, $\langle X, Y\rangle = \langle X^\Gamma, Y^\Gamma\rangle$. Thus

$$h_{\mathcal{X}}(\Phi\Phi^T) = h_{\mathcal{X}}((\Phi\Phi^T)^\Gamma) \leq \|(\Phi\Phi^T)^\Gamma\|_{2\to 2} = 1.$$

This last equality follows from the fact that $(\Phi\Phi^T)^\Gamma = F$.

Putting together these ingredients, we obtain the proof of the theorem. $\square$

It may seem surprising that the factor of $3^{1/4}$ emerges even for matrices with, say, ± 1 entries. An intuitive justification for this is that even if the columns of A are not Gaussian vectors, most linear combinations of them resemble Gaussians. The following Lemma shows that this behavior begins as soon as n is $\omega(1)$.

Lemma 15.3.4. *Let* $A = \sum_{i=1}^{m} e_i a_i^T/\sqrt{n}$ *with* $\mathbf{E}_i \|a_i\|_2^4 \geq 1$*. Then* $\|A\|_{2\to 4} \geq (3/(1+2/n))^{1/4}$.

To see that the denominator cannot be improved in general, observe that when $n = 1$ a random sign matrix will have $2 \to 4$ norm equal to 1.

Proof. Choose $x \in \mathbb{R}^n$ to be a random Gaussian vector such that $\mathbf{E}_x \|x\|_2^2 = 1$. Then

$$\mathop{\mathbf{E}}_x \|Ax\|_4^4 = \mathop{\mathbf{E}}_i \mathop{\mathbf{E}}_x n^{-2}(a_i^T x)^4 = n^2 \mathop{\mathbf{E}}_i \mathop{\mathbf{E}}_x \langle a_i, x\rangle^4 = 3 \mathop{\mathbf{E}}_i \|a_i\|_2^4 \geq 3. \tag{15.10}$$

The last equality comes from the fact that $\langle a_i, x\rangle$ is a Gaussian random variable with mean zero and variance $\|a_i\|_2^2/n$. On the other hand, $\mathbf{E}_x \|x\|_2^4 = 1 + 2/n$. Thus, there must exist an x for which $\|Ax\|_4^4/\|x\|_2^4 \geq 3/(1+2/n)$. □

Chapter 16

Hardness of $\textsc{Max}\Gamma$-2-$\textsc{Lin}$ and $\textsc{Max}\Gamma$-3-$\textsc{Lin}$ over integers

16.1 Introduction

In this chapter we consider one of the most fundamental algorithmic tasks: solving systems of linear equations. Given a ring R, the $\textsc{Max}k\text{-}\textsc{Lin}(R)$ problem is defined as follows: An input instance is a list of linear equations of the form $a_1x_{i_1} + \cdots + a_kx_{i_k} = b$, where $a_1, \ldots, a_k, b \in R$ are constants and $x_{i_1}, \ldots, x_{i_k}$ are variables from the set $\{x_1, \ldots, x_n\}$. Each equation also comes with a nonnegative rational weight; it is assumed the weights sum up to 1. The algorithmic task is to assign values from R to the variables so as to maximize the total weight of satisfied equations. We say that an assignment is γ-*good* if the equations it satisfies have total weight at least γ. We say that an algorithm achieves (c, s)-*approximation* if, whenever the instance has a c-good solution, the algorithm is guaranteed to find an s-good solution.

16.1.1 Prior work on $\textsc{Max}3\text{-}\textsc{Lin}(\mathbb{Z})$

It is an old result of Arora–Babai–Stern–Sweedyk [13] that for all $0 < \delta < 1$ there exists $\epsilon > 0$ and $k \in \mathbb{Z}^+$ such that it is NP-hard to $(\epsilon, \delta\epsilon)$-approximate $\textsc{Max}k\text{-}\textsc{Lin}(\mathbb{Q})$. Håstad's seminal work from 1997 [116] showed hardness even for very sparse, near-satisfiable instances: specifically, he showed that for all constant $\epsilon, \delta > 0$ and $q \in \mathbb{N}$, it is NP-hard to $(1 - \epsilon, 1/q + \delta)$-approximate $\textsc{Max}3\text{-}\textsc{Lin}(\mathbb{Z}_q)$. This is optimal in the sense that it is algorithmically easy to $(1, 1)$-approximate or $(c, 1/q)$-approximate $\textsc{Max}3\text{-}\textsc{Lin}(\mathbb{Z}_q)$. Håstad's hardness result even holds for the special case of MAXΓ-3-LIN($\mathbb{Z}_q$), meaning that all equations are of the form $x_{i_1} - x_{i_2} + x_{i_3} = b$.

Håstad's proof does not strictly generalize the ABSS [13] result on MAXk-LIN($\mathbb{Q}$) because there is no obvious reduction from hardness over $\mathbb{Z}_q$ to hardness over $\mathbb{Q}$. Indeed, it was not until much later, 2006, that NP-hardness of $(1-\epsilon, \delta)$-approximating MAXk-LIN($\mathbb{Q}$) was shown [92, 108]. Finally, in 2007 Guruswami and Raghavendra [109] generalized all of [13, 92, 108] by showing NP-hardness of $(1-\epsilon, \delta)$-approximating MAXΓ-3-LIN($\mathbb{Z}$). As we will see shortly, this easily implies the same hardness for MAXΓ-3-LIN($\mathbb{Q}$) and MAXΓ-3-LIN($\mathbb{R}$). Indeed, it shows a kind of "bicriteria" hardness: given a MAXΓ-3-LIN($\mathbb{Z}$) instance with a $(1-\epsilon)$-good solution over $\mathbb{Z}$, it is NP-hard to find a δ-good solution even over $\mathbb{R}$. Guruswami and Raghavendra's proof followed that of Håstad's to some extent, but involved somewhat technically intricate derandomized Long Code testing, using Fourier analysis with respect to a certain exponential distribution on $\mathbb{Z}^+$.

We would also like to mention the very recent work of Khot and Moshkovitz [139]. Motivated by proving the Unique Games Conjecture, they showed a strong NP-hardness result for a homogeneous variant of MAX3-LIN($\mathbb{R}$). Specifically, they considered the case where all equations are of the form $a_1 x_{i_1} + a_2 x_{i_2} + a_3 x_{i_3} = 0$ with $a_1, a_2, a_3 \in [\frac{1}{2}, 2]$. Very roughly speaking, they showed there is a universal $\delta > 0$ such that for all $\epsilon > 0$ the following problem is NP-hard: given an instance where there is a "Gaussian-distributed" real assignment which is $(1-\epsilon)$-good, find a Gaussian-distributed assignment in which the weight of equations satisfied to within margin $\delta\sqrt{\epsilon}$ is at least $1-\delta$. This result is incomparable to the one in [109].

16.1.2 Prior work on MAX2-LIN

Following Håstad's work there was five years of no progress on MAX2-LIN(R) for any ring R. To circumvent this, in 2002 Khot [136] introduced the Unique Games (UG) Conjecture, which would prove to be very influential (and notorious!). Khot showed a strong "UG-hardness" result for MAX2-LIN($\mathbb{Z}_2$) (crediting the result essentially to Håstad), namely that for all $t > 1/2$ and sufficiently small $\epsilon > 0$ it is UG-hard to $(1-\epsilon, 1-\epsilon^t)$-approximate MAX2-LIN($\mathbb{Z}_2$). This result is essentially optimal due to the Goemans–Williamson algorithm [94].

In 2004, Khot–Kindler–Mossel–O'Donnell [141] (using [175]) extended this work by showing that for all $\epsilon, \delta > 0$, there exists $q \in \mathbb{N}$ such that $(1-\epsilon, \delta)$-approximating MAXΓ-2-LIN($\mathbb{Z}_q$) is UG-hard, and hence in fact UG-*complete*. Here Γ-2-LIN means that all equations are of the form $x_{i_1} - x_{i_2} = b$. KKMO gave a quantitative dependence as well: given ϵ and q one can choose any $\delta > q\Lambda_{1-\epsilon}(1/q) \approx (1/q)^{\epsilon/(2-\epsilon)}$, where $\Lambda_{1-\epsilon}(1/q)$ is a certain correlated Gaussian quadrant probability.

The following natural question was left open by KKMO [141]:

Question 16.1.1. Is it true that for all $\epsilon, \delta > 0$ it is UG-hard to $(1-\epsilon, \delta)$-approximate MAXΓ-2-LIN($\mathbb{Z}$)?

The key technical tool used in the KKMO hardness result for MAX2-LIN($\mathbb{Z}_q$), namely the Majority Is Stablest Theorem [175], has a bad dependence on the parameter q. Thus pushing q to be "superconstantly" large seemed to pose a fundamental problem. The question above is one of the open problems posed at the end of Raghavendra's monumental thesis [190].

16.1.3 Our contributions

In this chapter we show that it is relatively easy to modify the proofs of the hardness results known for MAXΓ-2-LIN($\mathbb{Z}_q$) and MAXΓ-3-LIN($\mathbb{Z}_q$) to obtain $(1-\epsilon, \delta)$-approximation hardness results for MAXΓ-2-LIN($\mathbb{Z}$) and MAXΓ-3-LIN($\mathbb{Z}$). (Here Γ-3-LIN means that all equations are of the form $x_{i_1} + x_{i_2} - x_{i_3} = b$.) Thus we resolve the open question about MAXΓ-2-LIN($\mathbb{Z}$) and give a simpler proof of the Guruswami–Raghavendra [109] result. Our results also hold over $\mathbb{R}$ and over "superconstantly large" cyclic groups $\mathbb{Z}_q$ (we are not aware of previously known hardness results over $\mathbb{Z}_q$ when q is superconstant and prime). The results also have an essentially optimal quantitative tradeoff between ϵ, δ, and the magnitudes of the "right-hand side constants" b.

To state our two theorems, let us define B-BOUNDED-MAXΓ-2-LIN and B-BOUNDED-MAXΓ-3-LIN to be the special cases of MAXΓ-3-LIN and MAXΓ-2-LIN in which all right-hand side constants b are integers satisfying $|b| \le B$. Given an instance $\mathcal{I}$ of MAXΓ-k-LIN with integer constants b, we use the notation $\mathsf{opt}_R(\mathcal{I})$ to denote the maximum weight of equations that can be satisfied when the equations are evaluated over R.

Theorem 16.1.2. *For all constant $\epsilon, \gamma, \kappa > 0$ and constant $q \in \mathbb{N}$, given a q-BOUNDED-MAXΓ-2-LIN instance $\mathcal{I}$ it is UG-hard to distinguish the following two cases:*

- (Completeness.) *There is a $(1-\epsilon-3\gamma)$-good assignment over $\mathbb{Z}$; i.e., $\mathsf{opt}_{\mathbb{Z}}(\mathcal{I}) \ge 1-\epsilon-3\gamma$.*

- (Soundness.) *There is no* $(q\Lambda_{1-\epsilon}(1/q)+\kappa)$*-good assignment over* $\mathbb{Z}_q$*; i.e.,* $\mathsf{opt}_{\mathbb{Z}_q}(\mathcal{I}) \leq q\Lambda_{1-\epsilon}(1/q) + \kappa$.

Note that $q\Lambda_{1-\epsilon}(1/q) \approx (1/q)^{\epsilon/(2-\epsilon)}$ is the same soundness proved by KKMO [141] for MAXΓ-2-LIN($\mathbb{Z}_q$).

Theorem 16.1.3. *For all constant* $\epsilon, \kappa > 0$ *and* $q \in \mathbb{N}$*, given a* q-BOUNDED-MAXΓ-3-LIN *instance* $\mathcal{I}$ *it is* NP*-hard to distinguish the following two cases:*

- (Completeness.) *There is a* $(1 - O(\epsilon))$*-good assignment over* $\mathbb{Z}$*, i.e.,* $\mathsf{opt}_{\mathbb{Z}}(\mathcal{I}) \geq 1 - O(\epsilon)$.

- (Soundness.) *There is no* $(1/q + \kappa)$*-good assignment over* $\mathbb{Z}_q$*; i.e.,* $\mathsf{opt}_{\mathbb{Z}_q}(\mathcal{I}) \leq 1/q + \kappa$.

Note that $\mathsf{opt}_{\mathbb{Z}}(\mathcal{I}) \leq \mathsf{opt}_{\mathbb{Z}_q}(\mathcal{I})$ since we can convert a δ-good assignment over $\mathbb{Z}$ to a δ-good assignment over $\mathbb{Z}_q$ by reducing the integer solution modulo q. Therefore our hardness results are of the strongest "bicriteria" type: even when promised that there is near-perfect solution over $\mathbb{Z}$, it is hard for an algorithm to find a slightly good solution over $\mathbb{Z}_q$. Indeed, by virtue of Lemma 16.6.1 in Section 16.6, by losing just a constant factor in the soundness, we can show that it is also hard for an algorithm to find a slightly good solution over any ring $\{\mathbb{R}, \mathbb{Z}, \mathbb{Z}_{q+1}, Z_{q+2}, \ldots\}$ of the algorithm's choosing. Our results subsume and unify all aforementioned results on MAX3-LIN($\mathbb{Z}_q$), MAX3-LIN($\mathbb{Z}$), and MAX2-LIN($\mathbb{Z}_q$), and also provide an optimal UG-hardness result for MAXΓ-2-LIN($\mathbb{Z}$).

16.2 Preliminaries

16.2.1 Notations and Definitions

We write $\mathbb{Z}_q$ for the integers modulo q, and we identify the elements with $\{0, 1, \ldots, q-1\} \in \mathbb{Z}$. We sometimes write $\oplus_q$ for addition of integers modulo q and $+$ for addition over the integers. For two vectors $x, y \in \mathbb{Z}^n$, both $x \oplus_q y$ and $x + y$ are coordinate-wise add. We will also write Δ_q for the set of probability distributions over $\mathbb{Z}_q$. We can identify Δ_q with the standard $(q-1)$-dimensional simplex in $\mathbb{R}^q$. We also identify an element $a \in \mathbb{Z}_q$ with a distribution in Δ_q, namely, the distribution that puts all of its probability mass on a.

Fix $x \in \mathbb{Z}_q^n$, a random variable $\boldsymbol{y}$ is $(1-\epsilon)$-correlated to x, i.e. $\boldsymbol{y} \sim_{1-\epsilon} x$, if $\boldsymbol{y}$ can be get by rerandomizing each coordinate of x independently with probability ϵ.

We recall some standard definitions from the harmonic analysis of boolean functions (see, e.g., [190]). We will be considering functions of the form $f : \mathbb{Z}_q^n \to \mathbb{R}$. The set of all functions $f : \mathbb{Z}_q^n \to \mathbb{R}$ forms an inner product space with inner product

$$\langle f, g \rangle = \mathop{\mathbf{E}}_{\boldsymbol{x} \sim \mathbb{Z}_q^n}[f(\boldsymbol{x}) \cdot g(\boldsymbol{x})],$$

where $\boldsymbol{x} \sim \mathbb{Z}_q^n$ means that $\boldsymbol{x}$ is uniform randomly chosen from $\mathbb{Z}_q^n$. We also write $\|f\|_2 = \sqrt{\langle f, f \rangle}$ as usual.

The following Efron–Stein decomposition theorem is well-known; see [141].

Theorem 16.2.1. *Any $f : \mathbb{Z}_q^n \to \mathbb{R}$ can be uniquely decomposed as a sum of functions*

$$f(x) = \sum_{S \subseteq [n]} f^S(x),$$

where

- *$f^S(x)$ depends only on $x_S = (x_i, i \in S)$,*
- *for every $S \subseteq [n]$, for every S' such that $S \setminus S' \neq \emptyset$, and for every $y \in \mathbb{Z}_q^n$, it holds that*

$$\mathop{\mathbf{E}}_{\boldsymbol{x} \sim \mathbb{Z}_q^n}[f^S(\boldsymbol{x}) | \boldsymbol{x}_{S'} = y_{S'}] = 0.$$

Definition 16.2.2 (Influences)**.** *For functions $f : \mathbb{Z}_q^n \to \mathbb{R}$, define the influence of the i-th coordinate on f to be*

$$\mathbf{Inf}_i(f) = \sum_{S \ni i} \|f^S\|_2^2,$$

where $\|f^S\|_2^2 = \mathbf{E}_{\boldsymbol{x}}[f^S(\boldsymbol{x})^2]$. For functions $f : \mathbb{Z}_q^n \to \Delta_m$, let

$$\mathbf{Inf}_i(f) = \sum_{a \in \mathbb{Z}_m} \mathbf{Inf}_i(f_a),$$

where $f_a(x) = f(x)_a, \forall x \in \mathbb{Z}_q^n$.

Definition 16.2.3 (Noise operators)**.** *For functions $f : \mathbb{Z}_q^n \to \mathbb{R}$, define the noise operator $T_{1-\eta}$ to be*

$$T_{1-\eta} f(x) = \mathop{\mathbf{E}}_{\boldsymbol{y} \sim_{1-\eta} x}[f(\boldsymbol{y})].$$

For functions $f : \mathbb{Z}_q^n \to \Delta_m$, let $T_{1-\eta}$ be the noise operator so that $(T_{1-\eta} f)_a = T_{1-\eta}(f_a), \forall a \in \mathbb{Z}_q$.

Definition 16.2.4 (Noisy-influences)**.** *For functions $f : \mathbb{Z}_q^n \to \mathbb{R}$ and functions $f : \mathbb{Z}_q^n \to \Delta_m$, define the $(1-\eta)$-noisy-influence of the i-th coordinate of f to be*

$$\mathbf{Inf}_i^{(1-\eta)}(f) = \mathbf{Inf}_i(T_{1-\eta} f).$$

Fact 16.2.5. *For functions $f : \mathbb{Z}_q^n \to \Delta_m$, we have*

$$\sum_{i \in [n]} \mathbf{Inf}_i^{(1-\eta)}(f) \leq 1/\eta.$$

Proposition 16.2.6. *Let $f^{(1)}, \ldots, f^{(t)}$ be a collection of functions $\mathbb{Z}_q^n \to \mathbb{R}^m$. Then*

$$\mathbf{Inf}_i^{(1-\eta)} \left[\underset{k \in [t]}{\operatorname{avg}} \left\{ f^{(k)} \right\} \right] \leq \underset{k \in [t]}{\operatorname{avg}} \left\{ \mathbf{Inf}_i^{(1-\eta)}[f^{(k)}] \right\}.$$

Here for any $c_1, c_2, \ldots c_t \in \mathbb{R}$ (or $\mathbb{R}^m$), we use the notation $\operatorname{avg}(c_1, \ldots, c_t)$ to denote their average:

$$\frac{\sum_{i=1}^t c_i}{t}.$$

Both of Fact 16.2.5 and Proposition 16.2.6 are easy to verify by the definition of Noisy-influences. The proofs of the facts can be found in, e.g., [180] and [181].

Definition 16.2.7 (Noise stability)**.** *For functions $f : \mathbb{Z}_q^n \to \mathbb{R}$, define its stability against ϵ noise to be*

$$\mathbf{Stab}_{1-\epsilon}[f] = \underset{\boldsymbol{x} \sim \mathbb{Z}_q^n, \boldsymbol{y} \sim_{1-\epsilon} \boldsymbol{x}}{\mathbf{E}} [f(\boldsymbol{x}) f(\boldsymbol{y})].$$

One tool we need is the *Majority Is Stablest Theorem* from [175]. (We state here a version using a small noisy-influences assumption rather than a small "low-degree influences" assumption; see, e.g., Theorem 3.2 in [190] for a sketch of the small modification to [175] needed.)

Theorem 16.2.8. *For every function $f : \mathbb{Z}_q^n \to [0,1]$ such that $\mathbf{Inf}_i^{(1-\eta)}[f] \leq \tau$ for all $i \in [n]$, Let $\mu = \mathbf{E}[f]$. Then for any $0 < \epsilon < 1$,*

$$\mathbf{Stab}_{1-\epsilon}[f] \leq \Lambda_{1-\epsilon}(\mu) + e(\tau, q, \eta).$$

Here if we fix η, q, $e(\tau, \eta, q)$ goes to 0 when τ goes to 0.

In the above theorem, the quantity $\Lambda_{1-\epsilon}(\mu)$ is defined to be $\Pr[\boldsymbol{x}, \boldsymbol{y} \leq t]$ when $(\boldsymbol{x}, \boldsymbol{y})$ are joint standard Gaussians with covariance $1 - \epsilon$ and t is defined by $\Pr[\boldsymbol{x} \leq t] = \mu$.

16.3 Review of proofs of MAXΓ-2-LIN($\mathbb{Z}_q$) and MAXΓ-3-LIN($\mathbb{Z}_q$) hardness

As mentioned, we prove Theorem 16.1.2 and Theorem 16.1.3 by fairly easy modifications of the known hardness results for MAXΓ-2-LIN($\mathbb{Z}_q$) and MAXΓ-3-LIN($\mathbb{Z}_q$), due respectively to Khot–Kindler–Mossel–O'Donnell [141] and Håstad [116]. In this section, we review several places in the two proofs that are related to our modifications. We also assume the reader's familiarity with these works.

16.3.1 MAXΓ-2-LIN

Let us begin with MAXΓ-2-LIN. As shown in [141], to prove UG-hardness of $(1-\epsilon,\delta)$-approximating MAXΓ-2-LIN($\mathbb{Z}_q$) for constant κ and q, where $\delta = q\Lambda_{1-\epsilon}(1/q)+\kappa$, it suffices to construct a "Dictator vs. Small Low-Degree-Influences Test" (or, Dictator Test for short) for functions $f : \mathbb{Z}_q^L \to \Delta_q$ which uses Γ-2-LIN constraints and has completeness $1-\epsilon$, soundness δ. We recall the definition of Dictator Test as follows.

Generally speaking, a $1-\epsilon$ vs. δ Dictator Test for functions $f : \mathbb{Z}_q^L \to \mathbb{Z}_q$ is defined by a distribution over Γ-2-LIN constraints (over the entries of f). We say f passes the test when a random constraint (from the distribution) is satisfied by f. At the completeness side, all the L dictators (i.e., $f(x) = x_i$ for some $i \in L$) pass the test with probability at least $1-\epsilon$. At the soundness side, all functions with small noisy-influences (on all coordinates) pass the test with probability at most δ. KKMO indeed needs to construct a Dictator Test for functions of distributions, i.e., for $f : \mathbb{Z}_q^L \to \Delta_q$, where whenever the test refers an entry $f(x)$ for an element in $\mathbb{Z}_q$, it randomly samples an element from the distribution $f(x)$.

The Dictator Test used by KKMO is indeed a noise-stability test. Intuitively, dictator functions have high noise stability, while functions far from dictators have low noise stability. Note that this intuition is true only for balanced functions, as constant functions are far from dictators but very noise stable. Therefore, KKMO used the "folding" trick (which was introduced in [116]) to ensure that f outputs $1, 2, \ldots, q$ with the same probability.

16.3.2 MAXΓ-3-LIN

Let us move on to MAXΓ-3-LIN and our proof of Theorem 16.1.3. Håstad essentially showed that to prove NP-hardness of $(1-\epsilon, 1/q+\kappa)$-approximating MAXΓ-3-LIN($\mathbb{Z}_q$)

for constant q, it suffices to construct a "Matching-Dictator Test" on two functions for $f : \mathbb{Z}_q^K \to \mathbb{Z}_q$, $g : \mathbb{Z}_q^L \to \mathbb{Z}_q$ and $\pi : L \to K$. The test is defined by a distribution over $\boldsymbol{x} \in \mathbb{Z}_q^K, \boldsymbol{y} \in \mathbb{Z}_q^L, \boldsymbol{z} \in \mathbb{Z}_q^L$ with the check $f(x) + g(y) - g(z) = c \mod q$. Håstad's Test has the following completeness and soundness promises:

- If $f(x) = x_i$ and $g(y) = y_j$ such that $\pi(i) = j$, then f and g passes with probability $1 - \epsilon$.

- If f and g passes the test with probability $1/q + \kappa$, then there is a randomized procedure that "decodes" f into a coordinate $i \in L$ and g into a coordinate $j \in K$ such that $\pi(i) = j$ with constant probability depending only on q, ϵ, κ and independent of L, K, π. Also note that the decoding processes for f and g should be independent from each other.

Håstad constructed the following test: choose $\boldsymbol{x} \in \mathbb{Z}_q^K$ and $\boldsymbol{y} \in \mathbb{Z}_q^L$ uniformly and independently, define $\boldsymbol{z} \in \mathbb{Z}_q^L$ to be $\boldsymbol{z} = \boldsymbol{y} \oplus_q (\boldsymbol{x} \circ \pi)$, where $(\boldsymbol{x} \circ \pi)_i := \boldsymbol{x}_{\pi(i)}$, let $\boldsymbol{z}'$ be $(1 - \epsilon)$-correlated to $\boldsymbol{z}$, and test $f(\boldsymbol{x}) \oplus_q g(\boldsymbol{y}) = g(\boldsymbol{z}')$. Such a test does not work when $f \equiv 0$; thus Håstad introduced and used his method of folding (which was also used [141]) to ensure that f outputs $1, 2, \ldots, q$ with equal probability.

16.4 Overview of our proofs

As mentioned, we obtain Theorem 16.1.2 and Theorem 16.1.3 by modifying the KKMO [141] and Håstad [116] proofs. In this section we describe the idea of the modifications.

16.4.1 Active folding

The usual folding trick [116] enforces that f is balanced by replacing references to $f(x_1, \ldots, x_L)$ with references to $f(x_1 \oplus_q x_{j^*}, x_2 \oplus_q x_{j^*}, \ldots, x_L \oplus_q x_{j^*}) \oplus_q (-x_{j^*})$ for some arbitrary $j^* \in [L]$. (I.e., the reduction only uses q^{L-1} variables to represent f as opposed to q^L. Note that this makes the test's constraints of the form $f(x) \oplus_q b = f(x') \oplus_q b'$, but this is still of Γ-2-LIN type. We call this trick *static folding*.

Let us explain the alternative to "static folding" which we call *active folding*. Active folding is nothing more than building the folding directly into the test. We feel that this is slightly more natural than static folding, and as we will see it proves to be more flexible.

In the KKMO context of $\textsc{Max}\Gamma\text{-2-}\textsc{Lin}(\mathbb{Z}_q)$, active folding means that the test additionally chooses $\boldsymbol{c}, \boldsymbol{c}' \sim \mathbb{Z}_q$ uniformly and independently, and then it checks the Γ-2-$\textsc{Lin}$ constraint

$$f(\boldsymbol{x} \oplus_q (\boldsymbol{c}, \ldots \boldsymbol{c})) \oplus_q (-\boldsymbol{c}) = f(\boldsymbol{x}' \oplus_q (\boldsymbol{c}', \ldots \boldsymbol{c}')) \oplus_q (-\boldsymbol{c}')$$

rather than $f(\boldsymbol{x}) = f(\boldsymbol{x}')$. To analyze the KKMO test with active folding, first note that completeness does not change. As for the soundness analysis, given a function $f : \mathbb{Z}_q^L \to \Delta_q$ we introduce $\widetilde{f} : \mathbb{Z}_q^L \to \Delta_q$ defined by

$$\widetilde{f}(x)_a = \mathop{\mathbf{E}}_{\boldsymbol{c} \sim \mathbb{Z}_q} [f(x \oplus_q (\boldsymbol{c}, \ldots, \boldsymbol{c}))_{a \oplus_q \boldsymbol{c}}]. \tag{16.1}$$

Then the probability f satisfies the test with active folding is precisely the probability that $\widetilde{f}$ satisfies the $\widetilde{f}(\boldsymbol{x}) = \widetilde{f}(\boldsymbol{x}')$ test (in the sense of randomized functions), namely $\mathbf{Stab}_{1-\epsilon}[\widetilde{f}]$. We can now proceed with the KKMO analysis; the key is that we still have $\mathbf{E}[\widetilde{f}_a] = 1/q$ for all $a \in \mathbb{Z}_q$. To see this, take $Q = q$ in the following lemma:

Lemma 16.4.1. *Let $f : \mathbb{Z}_Q^L \to \Delta_q$ and suppose $\widetilde{f} : \mathbb{Z}_Q^L \to \Delta_q$ is defined as in* (16.1). *Then $\mathbf{E}[\widetilde{f}_a] = 1/q$ for all $a \in \mathbb{Z}_q$.*

Proof. We have

$$\mathbf{E}[\widetilde{f}_a] = \mathop{\mathbf{E}}_{\boldsymbol{x} \sim \mathbb{Z}_Q^K, \boldsymbol{c} \sim \mathbb{Z}_q} \left[f(\boldsymbol{x} \oplus_Q (\boldsymbol{c}, \ldots, \boldsymbol{c}))_{a \oplus_q \boldsymbol{c}} \right].$$

Write $\widetilde{\boldsymbol{x}} = \boldsymbol{x} \oplus_Q (\boldsymbol{c}, \ldots, \boldsymbol{c}) \in \mathbb{Z}_Q^K$. The distribution of $\widetilde{\boldsymbol{x}} \mid (\boldsymbol{c} = c)$ is uniform on $\mathbb{Z}_Q^K$ for every c. In other words, $\widetilde{\boldsymbol{x}}$ and $\boldsymbol{c}$ are independent. Thus

$$\mathbf{E}[\widetilde{f}_a] = \mathop{\mathbf{E}}_{\widetilde{\boldsymbol{x}}} \left[\mathop{\mathbf{E}}_{\boldsymbol{c}} \left[f(\widetilde{\boldsymbol{x}})_{a \oplus_q \boldsymbol{c}} \right] \right] = \mathop{\mathbf{E}}_{\widetilde{\boldsymbol{x}}} \left[(1/q) \sum_{b \in \mathbb{Z}_q} [f(\widetilde{\boldsymbol{x}})_b] \right] = \mathop{\mathbf{E}}_{\widetilde{\boldsymbol{x}}}[1/q] = 1/q. \quad \square$$

16.4.2 Modifying the KKMO proof

We now describe how to obtain Theorem 16.1.2. Let us first ask: Why does the KKMO reduction (with active folding) not prove Theorem 16.1.2 already? The soundness statement of Theorem 16.1.2 would hold since it is over $\mathbb{Z}_q$. The problem is in the completeness statement: a dictator $f : \mathbb{Z}_q^L \to \mathbb{Z}$, $f(x) = x_i$ does not satisfy the the KKMO test with probability close to 1. The reason is that folding may introduce *wrap-around* in $\mathbb{Z}_q$. More specifically (and ignoring the ϵ noise), the KKMO test with active folding will check

$$(\boldsymbol{x}_i + \boldsymbol{c} \bmod q) - \boldsymbol{c} = (\boldsymbol{x}_i + \boldsymbol{c}' \bmod q) - \boldsymbol{c}' \tag{16.2}$$

over the integers, and this is only satisfied if both $\boldsymbol{x}_i + \boldsymbol{c}$ and $\boldsymbol{x}_i + \boldsymbol{c}'$ wrap around, or neither does: probability $1/2$. (The situation with static folding is similar.)

Sketch of a first fix. There is a simple way to somewhat fix the completeness: choose $\boldsymbol{c}$ and $\boldsymbol{c}'$ from a range smaller than $\{0, 1, \ldots, q-1\}$. E.g., if we choose $\boldsymbol{c}$ and $\boldsymbol{c}'$ independently and uniformly in $\{0, 1, \ldots, \lfloor q/t \rfloor\}$, then we get wrap-around in $\boldsymbol{x}_i + \boldsymbol{c}$ with probability at most $1/t$. Hence the dictator $f(x) = x_i$ will satisfy the test (16.2) over $\mathbb{Z}$ with probability at least $1 - 2/t$, which we can make close to 1 by taking t large. Now how does this restricted folding affect the soundness analysis? If we redefine the folded function $\widetilde{f}$ appropriately, it is not hard to show that we will have $\mathbf{E}[\widetilde{f}_a] \leq (t/q)$ for all a. We could then proceed with the KKMO analysis applied to $\widetilde{f}$ and obtain soundness $q\Lambda_{1-\epsilon}(t/q)$. Choosing, say, $t = \log q$ would achieve a good completeness versus soundness tradeoff; roughly $1 - \epsilon'$ versus $\widetilde{O}(1/q)^{\epsilon/(2-\epsilon)}$.

A better fix. A slight twist on this idea actually gives the optimal completeness versus soundness tradeoff. Instead of restricting the range of the folding, we simply enlarge the domain of f. Specifically, let $\gamma > 0$ be any small constant and define $Q = \lceil q/\gamma \rceil$. To prove Theorem 16.1.2 we run the KKMO reduction with functions f whose domain is $\mathbb{Z}_Q^L$. We still active folding with $\boldsymbol{c} \in \mathbb{Z}_q$. In other words, the test chooses $\boldsymbol{x}, \boldsymbol{x}'$ to be $(1-\epsilon)$-correlated strings in $\mathbb{Z}_Q^L$, chooses $\boldsymbol{c}, \boldsymbol{c}' \in \mathbb{Z}_q$ uniformly and independently, and outputs the constraint $f(\boldsymbol{x} \oplus_Q (\boldsymbol{c}, \ldots, \boldsymbol{c})) - \boldsymbol{c} = f(\boldsymbol{x}' \oplus_Q (\boldsymbol{c}', \ldots, \boldsymbol{c}')) - \boldsymbol{c}'$. Note that this is a q-BOUNDED-Γ-2-LIN constraint. As the 'wrap-around" probability is $q/Q \leq \gamma$, we have completeness over $\mathbb{Z}$ of at least $1 - \epsilon - \gamma$. As for the soundness over $\mathbb{Z}_q$, we now need to consider functions $f : \mathbb{Z}_Q^L \to \Delta_q$. If we introduce the folded function $\widetilde{f} : \mathbb{Z}_Q^L \to \Delta_q$ as in (16.1), the probability f passes the test over $\mathbb{Z}_q$ is again $\mathbf{Stab}_{1-\epsilon}[\widetilde{f}]$, and we still have $\mathbf{E}[\widetilde{f}_a] = 1/q$ by Lemma 16.4.1. Hence the soundness analysis for Theorem 16.1.2 becomes essentially identical to the soundness analysis for KKMO with active folding. The only tiny difference is that we need to apply the Majority Is Stablest Theorem with domain $\mathbb{Z}_Q^L$ rather than $\mathbb{Z}_q^L$. But Q is still a constant since γ and q are; hence we obtain the claimed $1 - \epsilon - \gamma$ completeness over $\mathbb{Z}$ and $q\Lambda_{1-\epsilon}(1/q)$ soundness over $\mathbb{Z}_q$.

16.4.3 Modifying the Håstad proof

The modification to Håstad's test needed to obtain Theorem 16.1.3 is similar. If one carries out Håstad's proof using the Efron–Stein decomposition rather than harmonic analysis over $\mathbb{Z}_q$, one sees that the soundness relies entirely on $\mathbf{E}[f_a] = 1/q$ for all $a \in \mathbb{Z}_q$. Thus we only need to apply folding to f. Let us examine the Håstad Γ-3-LIN test on $f : \mathbb{Z}_q^K \to \mathbb{Z}_q$, $g : \mathbb{Z}_q^L \to \mathbb{Z}_q$, and $\pi : L \to K$. We will use active folding on f, and for simplicity of this discussion ignore the ϵ-noise. The test chooses $\boldsymbol{x} \sim \mathbb{Z}_q^K$ and $\boldsymbol{y} \sim \mathbb{Z}_q^L$ uniformly and

Figure 16.1: Test $\mathcal{T}$ with parameters ϵ, γ, q for functions on $\mathbb{Z}_Q^K$

- Choose $\boldsymbol{x}, \boldsymbol{x}' \sim \mathbb{Z}_Q^K$ to be a pair of $(1-\epsilon)$-correlated random strings.
- Choose $\boldsymbol{c}, \boldsymbol{c}' \sim [q]$ independently and uniformly.
- Define $\widetilde{\boldsymbol{x}} = \boldsymbol{x} \oplus_Q (\boldsymbol{c}, \boldsymbol{c}, \dots, \boldsymbol{c})$, and define $\widetilde{\boldsymbol{x}}' = \boldsymbol{x}' \oplus_Q (\boldsymbol{c}', \boldsymbol{c}', \dots, \boldsymbol{c}')$.
- Test the constraint $f(\widetilde{\boldsymbol{x}}) - \boldsymbol{c} = f(\widetilde{\boldsymbol{x}}') - \boldsymbol{c}'$.

independently, defines $\boldsymbol{z} \in \mathbb{Z}_q^L$ by $\boldsymbol{z} = \boldsymbol{y} \oplus_q (\boldsymbol{x} \circ \pi)$ (again, $(\boldsymbol{x} \circ \pi)_i := \boldsymbol{x}_{\pi(i)}$), chooses $\boldsymbol{c} \sim \mathbb{Z}_q$ uniformly, and finally checks the Γ-3-LIN constraint

$$f(\boldsymbol{x} \oplus_q (\boldsymbol{c}, \dots, \boldsymbol{c})) - \boldsymbol{c} + g(\boldsymbol{y}) = g(\boldsymbol{z}).$$

Again, if we simply use this reduction in an attempt to prove Theorem 16.1.3, the soundness is fine but the completeness over $\mathbb{Z}$ is a problem due to wrap-around. Indeed, there are two possibilities for wrap-around here: in $\boldsymbol{x}_i + \boldsymbol{c}$ and in $\boldsymbol{y}_j + \boldsymbol{x}_{\pi(j)}$. We mitigate this with the same idea used for MAXΓ-2-LIN. Given constants ϵ and q we define constants $Q = \lceil q/\epsilon \rceil$ and $\mathcal{Q} = \lceil Q/\epsilon \rceil$. We enlarge f's domain to $\mathbb{Z}_Q^K$ and g's domain to $\mathbb{Z}_{\mathcal{Q}}^L$. We continue to fold f using $\boldsymbol{c} \sim \mathbb{Z}_q$. Now the two possibilities for wrap-around occur with probability at most ϵ each and hence the completeness over $\mathbb{Z}$ is $1 - O(\epsilon)$. Defining $\widetilde{f} : \mathbb{Z}_Q^K \to \Delta_q$ as in (16.1), we again have $\mathbf{E}[\widetilde{f}_a] = 1/q$ for each $a \in \mathbb{Z}_q$ and can carry out the (Efron–Stein-style) Håstad soundness analysis, obtaining soundness $1/q + \kappa$ over $\mathbb{Z}_q$.

16.5 Dictator Test details

16.5.1 Dictator Test for MAXΓ-2-LIN

Given constants $\epsilon, \gamma, \kappa > 0$ and $q, K \in \mathbb{Z}^+$, let $Q = \lceil q/\gamma \rceil$. We define the Dictator Test $\mathcal{T}$ for functions f with domain $\mathbb{Z}_Q^K$ as in Figure 16.1. Let $\mathsf{val}_{\mathbb{Z}}^{\mathcal{T}}(f)$ be the probability that f passes the test, and let $\mathsf{val}_{\mathbb{Z}_q}^{\mathcal{T}}(f)$ be the probability that f passes the test over $\mathbb{Z}_q$.

Theorem 16.5.1. *There exists $\tau, \eta > 0$ such that $\mathcal{T}$ is a q-BOUNDED-Γ-2-LIN test with following properties:*

- (Completeness.) *Each of the K dictators $f : \mathbb{Z}_Q^K \to \mathbb{Z}$ has* $\mathsf{val}_{\mathbb{Z}}^{\mathcal{T}}(f) \geq 1 - \epsilon - \gamma$.
- (Soundness.) *Let $f : \mathbb{Z}_Q^K \to \Delta_q$ and define $\widetilde{f} : \mathbb{Z}_Q^K \to \Delta_q$ as in* (16.1). *Suppose that $\mathbf{Inf}_i^{(1-\eta)}[\widetilde{f}] \leq \tau$ for all $i \in [K]$. Then $\mathsf{val}_{\mathbb{Z}_q}^{\mathcal{T}}(f) \leq q\Lambda_{1-\epsilon}(1/q) + \kappa$, where $\kappa = \kappa(\tau, Q, \eta) > 0$ can be made arbitrarily small by taking $\tau, \eta > 0$ sufficiently small.*

Theorem 16.5.1 together with the following lemma proves Theorem 16.1.2.

Lemma 16.5.2. *Theorem 16.5.1 implies Theorem 16.1.2 .*

Lemma 16.5.2 is implicit from [141], and is proved in Section 16.7.1.

Proof of Theorem 16.5.1. For the Completeness case, we need to analyze for a fixed $i \in [K]$ the probability that

$$(\boldsymbol{x}_i \oplus_Q \boldsymbol{c}) - \boldsymbol{c} = (\boldsymbol{x}_i' \oplus_Q \boldsymbol{c}') - \boldsymbol{c}' \tag{16.3}$$

holds over $\mathbb{Z}$. We have $\boldsymbol{x}_i = \boldsymbol{x}_i'$ except with probability at most ϵ, and $\boldsymbol{x}_i \leq Q - q$ except with probability at most $q/Q \leq \gamma$. When both of these events occur, equation (16.3) holds. This proves the completeness.

As for the Soundness case, by Lemma 16.4.1 we have $\mu_a = \mathbf{E}[\widetilde{f}_a] = 1/q$ for each $a \in \mathbb{Z}_q$. By assumption we have $\mathbf{Inf}_i^{(1-\eta)}[\widetilde{f}_a] \leq \mathbf{Inf}_i^{(1-\eta)}[\widetilde{f}] \leq \tau$. Thus from Theorem 16.2.8 we obtain $\mathbf{Stab}_{1-\epsilon}[\widetilde{f}_a] \leq \Lambda_{1-\epsilon}(1/q) + e(\tau, Q, \eta)$ for each a. Summing this over $a \in \mathbb{Z}_q$ yields

$$\mathbf{Stab}_{1-\epsilon}[\widetilde{f}] \leq q\Lambda_{1-\epsilon}(1/q) + q \cdot e(\tau, Q, \eta).$$

The proof is completed by taking $\kappa = q \cdot e(\tau, Q, \eta)$, since $\mathbf{Stab}_{1-\epsilon}[\widetilde{f}] = \mathsf{val}_{\mathbb{Z}_q}^{\mathcal{T}}(f)$ by unrolling definitions. □

16.5.2 Matching Dictator Test for MAXΓ-3-LIN

Given constants $\epsilon, \kappa > 0$ and $q, L, K \in \mathbb{Z}$, let $Q = \lceil q/\epsilon \rceil$ and $\mathcal{Q} = \lceil Q/\epsilon \rceil$. In Figure 16.2, we define the Matching Dictator Test $\mathcal{U}$ for function f with domain $\mathbb{Z}_Q^K$, function g with domain $\mathbb{Z}_{\mathcal{Q}}^L$, and projection $\pi : L \to K$. Let $\mathsf{val}_{\mathbb{Z}}^{\mathcal{U}}(f, g)$ be the probability that f, g pass the test, and let $\mathsf{val}_{\mathbb{Z}_q}^{\mathcal{U}}(f, g)$ be the probability that f, g pass the test over $\mathbb{Z}_q$.

Theorem 16.5.3. *$\mathcal{U}$ is a q-BOUNDED-Γ3-LIN test satisfying:*

Figure 16.2: Test $\mathcal{U}$ with parameters ϵ, q for f on $\mathbb{Z}_Q^K$, g on $\mathbb{Z}_Q^L$, $\pi : L \to K$

- Choose $\boldsymbol{x} \sim \mathbb{Z}_Q^K$, $\boldsymbol{y} \sim \mathbb{Z}_Q^L$ uniformly and independently.
- Define $\boldsymbol{z} \in \mathbb{Z}_Q^L$ by $\boldsymbol{z} = \boldsymbol{y} \oplus_Q (\boldsymbol{x} \circ \pi)$.
- Choose $\boldsymbol{c} \sim \mathbb{Z}_q$ uniformly and define $\widetilde{\boldsymbol{x}} \in \mathbb{Z}_Q^K$ by $\widetilde{\boldsymbol{x}} = \boldsymbol{x} \oplus_Q (\boldsymbol{c}, \boldsymbol{c}, \ldots, \boldsymbol{c})$.
- Let $\boldsymbol{x}' \in \mathbb{Z}_Q^K$ be $(1-\epsilon)$-correlated to $\widetilde{\boldsymbol{x}}$, let $\boldsymbol{y}' \in \mathbb{Z}_Q^L$ be $(1-\epsilon)$-correlated to $\boldsymbol{y}$, and let $\boldsymbol{z}' \in \mathbb{Z}_Q^L$ be $(1-\epsilon)$-correlated to $\boldsymbol{z}$.
- Test the constraint $f(\boldsymbol{x}') - \boldsymbol{c} + g(\boldsymbol{y}') = g(\boldsymbol{z}')$.

- (Completeness.) *If $f : \mathbb{Z}_Q^K \to \mathbb{Z}$ and $g : \mathbb{Z}_Q^L \to \mathbb{Z}$ are matching dictators — i.e., $f(x) = x_{\pi(j)}$ and $g(y) = y_j$ for some $j \in [L]$ — then $\mathsf{val}_{\mathbb{Z}}^{\mathcal{U}}(f, g) \geq 1 - 5\epsilon$.*
- (Soundness.) *Let $f : \mathbb{Z}_Q^K \to \mathbb{Z}_q$, $g : \mathbb{Z}_Q^L \to \mathbb{Z}_q$ and define $\widetilde{f} : \mathbb{Z}_Q^K \to \Delta_q$ as in* (16.1). *Suppose that $\mathsf{val}_{\mathbb{Z}_q}^{\mathcal{U}}(f, g) \geq 1/q + \kappa$, then there is a randomized "decoding procedure" $\mathcal{D}$ which decodes g to a coordinate $\mathcal{D}(g) \in [L]$ and f to a coordinate $\mathcal{D}(f) \in [K]$ such that $\pi(\mathcal{D}(g)) = \mathcal{D}(f)$ with at least a constant probability $\zeta = \zeta(q, \epsilon, \kappa)$ independent of π, L, K.*

Theorem 16.5.3 together with the following lemma proves Theorem 16.1.3.

Lemma 16.5.4. *Theorem 16.5.3 implies Theorem 16.1.3 .*

Lemma 16.5.4 is proved in Section 16.7.2.

Proof of Theorem 16.5.3. Define $\widetilde{f} : \mathbb{Z}_Q^K \to \Delta_q$ as in (16.1). For the completeness case, we need to analyze for a fixed $j \in [L]$ the probability that

$$\boldsymbol{x}'_{\pi(j)} - \boldsymbol{c} + \boldsymbol{y}'_j = \boldsymbol{z}'_j \tag{16.4}$$

holds over $\mathbb{Z}$. Except with probability at most 3ϵ we have all of

$$\boldsymbol{x}'_{\pi(j)} = \widetilde{\boldsymbol{x}}_{\pi(j)} = \boldsymbol{x}_{\pi(j)} \oplus_Q \boldsymbol{c},$$
$$\boldsymbol{y}'_j = \boldsymbol{y}_j, \qquad \boldsymbol{z}'_j = \boldsymbol{z}_j = \boldsymbol{x}_{\pi(j)} \oplus_Q \boldsymbol{y}_j.$$

Except with probability at most $q/Q \le \epsilon$ we have $\boldsymbol{x}_{\pi(j)} \le Q - q$, in which case $\boldsymbol{x}_{\pi(j)} \oplus_Q \boldsymbol{c}$ equals $\boldsymbol{x}_{\pi(j)} + \boldsymbol{c}$. Except with probability at most $Q/\mathcal{Q} \le \epsilon$ we have $\boldsymbol{y}_j \le \mathcal{Q} - Q$, in which case $\boldsymbol{x}_{\pi(j)} \oplus_{\mathcal{Q}} \boldsymbol{y}_j = \boldsymbol{x}_{\pi(j)} + \boldsymbol{y}_j$. Thus when all five events occur, equation (16.4) indeed holds over $\mathbb{Z}$.

As for the soundness case, write $f' = T_{1-\epsilon}\widetilde{f}$ and $g' = T_{1-\epsilon}g$, where we think of g as $g : \mathbb{Z}_{\mathcal{Q}}^L \to \Delta_q$. By unrolling definitions we have

$$\mathsf{val}^{\mathcal{U}}_{\mathbb{Z}_q}(f, g) = \sum_{a,b \in \mathbb{Z}_q} \mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}[f'_a(\boldsymbol{x}) g'_b(\boldsymbol{y}) g'_{a \oplus_q b}(\boldsymbol{z})].$$

Write $\mu_a = \mathbf{E}[f'_a(\boldsymbol{x})]$. Thus $\mu_a = \mathbf{E}[\widetilde{f}_a] = 1/q$, by Lemma 16.4.1. We conclude that

$$\mathsf{val}^{\mathcal{U}}_{\mathbb{Z}_q}(f, g) = \sum_{a,b \in \mathbb{Z}_q} \mathbf{E}[(f'_a(\boldsymbol{x}) - \mu_a) g'_b(\boldsymbol{y}) g'_{a \oplus_q b}(\boldsymbol{z})] + (1/q) \sum_{a,b \in \mathbb{Z}_q} \mathbf{E}[g'_b(\boldsymbol{y}) g'_{a \oplus_q b}(\boldsymbol{z})].$$

The second term above is

$$(1/q) \sum_{a,b \in \mathbb{Z}_q} \mathbf{E}[g'_b(\boldsymbol{y}) g'_{a \oplus_q b}(\boldsymbol{z})] = (1/q)\, \mathbf{E}\big[(\sum_c g'_c(\boldsymbol{y})) \cdot (\sum_c g'_c(\boldsymbol{z}))\big] = (1/q)\, \mathbf{E}[1 \cdot 1] = 1/q,$$

since g' is Δ_q-valued. Thus to complete the proof it remains to show that if

$$\sum_{a,b \in \mathbb{Z}_q} \mathbf{E}[(f'_a(\boldsymbol{x}) - \mu_a) g'_b(\boldsymbol{y}) g'_{a \oplus_q b}(\boldsymbol{z})] \tag{16.5}$$

is at least $\kappa > 0$ then we can suitably decode $\widetilde{f}$ and g. Let us now apply the Efron–Stein decomposition to f' and g' with respect to the uniform distributions on their domains. Given $S \subseteq [K]$, $T \subseteq [L]$, for simplicity we write

$$\boldsymbol{F}_a^S = {f'}_a^S(\boldsymbol{x}), \;\; \boldsymbol{G}_b^T = {g'}_b^T(\boldsymbol{y}), \;\; \boldsymbol{H}_{a+b}^T = {g'}_{a \oplus_q b}^T(\boldsymbol{z}).$$

Thus

$$(16.5) = \sum_{a,b \in \mathbb{Z}_q} \mathbf{E}\Big[\Big(\sum_{\emptyset \ne S \subseteq [K]} \boldsymbol{F}_a^S\Big)\Big(\sum_{T \subseteq [L]} \boldsymbol{G}_b^T\Big)\Big(\sum_{U \subseteq [L]} \boldsymbol{H}_b^U\Big)\Big] = \sum_{a,b \in \mathbb{Z}_q} \sum_{\substack{\emptyset \ne S \subseteq [K] \\ T,U \subseteq [L]}} \mathbf{E}[\boldsymbol{F}_a^S \boldsymbol{G}_b^T \boldsymbol{H}_{a+b}^U].$$

Let us simplify the above. We have $\mathbf{E}[\boldsymbol{F}_a^S \boldsymbol{G}_b^T \boldsymbol{H}_{a+b}^U] = \mathbf{E}[\boldsymbol{F}_a^S \cdot \mathbf{E}[\boldsymbol{G}_b^T \boldsymbol{H}_{a+b}^U \mid \boldsymbol{x}]]$. Note that even if we condition on $\boldsymbol{x}$, the marginals on $\boldsymbol{y}$ and $\boldsymbol{z}$ are uniform on $\mathbb{Z}_{\mathcal{Q}}^L$. It follows from the properties of the Efron–Stein decomposition that $\mathbf{E}[\boldsymbol{G}_b^T \boldsymbol{H}_{a+b}^U \mid \boldsymbol{x}]$ is always 0 if $T \neq U$. Thus

$$(16.5) = \sum_{a,b \in \mathbb{Z}_q} \sum_{\substack{\emptyset \neq S \subseteq [K] \\ U \subseteq [L]}} \mathbf{E}[\boldsymbol{F}_a^S \boldsymbol{G}_b^U \boldsymbol{H}_{a+b}^U].$$

Similarly, conditioned on the U-coordinates of $\boldsymbol{y}$ and $\boldsymbol{z}$, the coordinates of $\boldsymbol{x}$ outside $\pi(U)$ are independent and uniform on $\mathbb{Z}_Q$. Hence $\mathbf{E}[\boldsymbol{F}_a^S \boldsymbol{G}_b^U \boldsymbol{H}_{a+b}^U] = 0$ if $S \not\subseteq \pi(U)$. We conclude that

$$(16.5) = \sum_{a,b \in \mathbb{Z}_q} \sum_{\substack{U \neq \emptyset \\ \emptyset \neq S \subseteq \pi(U)}} \mathbf{E}[\boldsymbol{F}_a^S \boldsymbol{G}_b^U \boldsymbol{H}_{a+b}^U] = \sum_{a,b \in \mathbb{Z}_q} \sum_{U \neq \emptyset} \mathbf{E}[\boldsymbol{F}_a^{\leq \pi(U)} \boldsymbol{G}_b^U \boldsymbol{H}_{a+b}^U],$$

where we defined $\boldsymbol{F}_a^{\leq \pi(U)} = \sum_{\emptyset \neq S \subseteq \pi(U)} \boldsymbol{F}_a^S$. Shifting the sum over a and b to the inside we obtain

$$(16.5) = \sum_{U \neq \emptyset} \mathbf{E}\Big[\sum_{a,b \in \mathbb{Z}_q} \boldsymbol{F}_a^{\leq \pi(U)} \boldsymbol{G}_b^U \boldsymbol{H}_{a+b}^U\Big] \leq \sum_{U \neq \emptyset} \mathbf{E}\left[\sqrt{\sum_{a,b} (\boldsymbol{F}_a^{\leq \pi(U)})^2 (\boldsymbol{G}_b^U)^2} \sqrt{\sum_{a,b} (\boldsymbol{H}_{a+b}^U)^2}\right],$$

having used Cauchy-Schwarz. We can think of, e.g., $(\boldsymbol{G}_0^U, \dots, \boldsymbol{G}_{q-1}^U)$ as a vector in $\mathbb{R}^q$; writing $\|\boldsymbol{G}^U\|$ for the Euclidean length of this vector (and similarly for $\boldsymbol{F}$ and $\boldsymbol{H}$), the right side above is precisely $\sqrt{q} \sum_{U \neq \emptyset} \mathbf{E}\big[\|\boldsymbol{F}^{\leq \pi(U)}\| \cdot \|\boldsymbol{G}^U\| \cdot \|\boldsymbol{H}^U\|\big]$. Thus

$$(16.5) \leq \sqrt{q} \sum_{U \neq \emptyset} \mathbf{E}\big[\|\boldsymbol{F}^{\leq \pi(U)}\| \cdot \|\boldsymbol{G}^U\| \cdot \|\boldsymbol{H}^U\|\big] \leq \sqrt{q} \sum_{U \neq \emptyset} \sqrt{\mathbf{E}[\|\boldsymbol{F}^{\leq \pi(U)}\|^2 \|\boldsymbol{G}^U\|^2]} \sqrt{\mathbf{E}[\|\boldsymbol{H}^U\|^2]},$$

using Cauchy-Schwarz again. Now $\boldsymbol{F}^{\leq \pi(U)}$ depends only on $\boldsymbol{x}$ and $\boldsymbol{G}^U$ depends only on $\boldsymbol{y}$; hence they are independent. Further, since $\boldsymbol{y}$ and $\boldsymbol{z}$ have the same distribution (though they are not independent), the same is true of $\boldsymbol{G}^U$ and $\boldsymbol{H}^U$. Hence

$$\begin{aligned}(16.5) &\leq \sqrt{q} \sum_{U \neq \emptyset} \sqrt{\mathbf{E}[\|\boldsymbol{F}^{\leq \pi(U)}\|^2]}\, \mathbf{E}[\|\boldsymbol{G}^U\|^2] \\ &\leq \sqrt{q} \sqrt{\sum_{U \neq \emptyset} \mathbf{E}[\|\boldsymbol{F}^{\leq \pi(U)}\|^2]\, \mathbf{E}[\|\boldsymbol{G}^U\|^2]} \sqrt{\sum_{U \neq \emptyset} \mathbf{E}[\|\boldsymbol{G}^U\|^2]},\end{aligned}$$

using Cauchy-Schwarz again. By (generalized) Parseval, $\sum_{U \neq \emptyset} \mathbf{E}[\|\boldsymbol{G}^U\|^2] \leq \sum_U \mathbf{E}[\|\boldsymbol{G}^U\|^2] = \mathbf{E}[\|\boldsymbol{G}\|^2] \leq 1$, since $\boldsymbol{G}$ takes values in Δ_q. Thus we finally conclude

$$(16.5) \leq \sqrt{q} \sqrt{\sum_{U \neq \emptyset} \mathbf{E}[\|\boldsymbol{F}^{\leq \pi(U)}\|^2]\, \mathbf{E}[\|\boldsymbol{G}^U\|^2]}$$

$$= \sqrt{q \sum_{U \neq \emptyset} \mathbf{E}[\|(T_{1-\epsilon}\widetilde{f})^{\leq \pi(U)}(\boldsymbol{x})\|^2]\, \mathbf{E}[\|(T_{1-\epsilon}g)^U(\boldsymbol{y})\|^2]}.$$

If $\mathsf{val}^{\mathcal{U}}_{\mathbb{Z}_q}(f, g) \geq 1/q + \kappa$, then we have $\kappa \leq$ (16.5) and therefore

$$\sum_{U \neq \emptyset} \mathbf{E}[\|(T_{1-\epsilon}\widetilde{f})^{\leq \pi(U)}(\boldsymbol{x})\|^2]\, \mathbf{E}[\|(T_{1-\epsilon}g)^U(\boldsymbol{y})\|^2] \geq \kappa^2/q. \tag{16.6}$$

We now define the decoding procedure. It works in a similar way as in Håstad's work [116], as follows. We sample a random set $S \subseteq [K]$ according to distribution $\mathbf{E}[\|\widetilde{f}^S(\boldsymbol{x})\|^2]$, and let $\mathcal{D}(f) \in S$ uniformly (or an arbitrary element of $[K]$ if $S = \emptyset$). We also sample a random set $T \subseteq [L]$ according to distribution $\mathbf{E}[\|g^T(\boldsymbol{y})\|^2]$, and choose $\mathcal{D}(g) \in T$ uniformly (or an arbitrary element of $[L]$ if $T = \emptyset$). We have

$$\begin{aligned}
&\Pr[\pi(\mathcal{D}(g)) = \mathcal{D}(f)] \\
\geq &\sum_{T, \emptyset \neq S \subseteq \pi(T)} \mathbf{E}[\|\widetilde{f}^S(\boldsymbol{x})\|^2]\, \mathbf{E}[\|g^T(\boldsymbol{y})\|^2] \frac{1}{|S|} \\
\geq &2\epsilon \sum_{T, \emptyset \neq S \subseteq \pi(T)} \mathbf{E}[\|\widetilde{f}^S(\boldsymbol{x})\|^2]\, \mathbf{E}[\|g^T(\boldsymbol{y})\|^2](1-\epsilon)^{2|S|},
\end{aligned}$$

where in the last step we use the fact $1/|S| \geq 2\epsilon(1-\epsilon)^{2|S|}$. Note that $\mathbf{E}[\|\widetilde{f}^S(\boldsymbol{x})\|^2](1-\epsilon)^{2|S|} = \mathbf{E}[\|T_{1-\epsilon}\widetilde{f}^S(\boldsymbol{x})\|^2]$ and $\mathbf{E}[\|g^T(\boldsymbol{y})\|^2] \geq \mathbf{E}[\|(T_{1-\epsilon}g)^T(\boldsymbol{y})\|^2]$, we have

$$\begin{aligned}
&\Pr[\pi(\mathcal{D}(g)) = \mathcal{D}(f)] \\
\geq &2\epsilon \sum_{T, \emptyset \neq S \subseteq \pi(T)} \mathbf{E}[\|(T_{1-\epsilon}\widetilde{f})^S(\boldsymbol{x})\|^2]\, \mathbf{E}[\|(T_{1-\epsilon}g)^T(\boldsymbol{y})\|^2] \\
= &2\epsilon \sum_{T} \mathbf{E}[\|(T_{1-\epsilon}\widetilde{f})^{\leq \pi(T)}(\boldsymbol{x})\|^2]\, \mathbf{E}[\|(T_{1-\epsilon}g)^T(\boldsymbol{y})\|^2] \\
\geq &2\epsilon\kappa^2/q,
\end{aligned}$$

where the second last step is by definition and orthogonality of $(T_{1-\epsilon}\widetilde{f})^{S_1}$ and $(T_{1-\epsilon}\widetilde{f})^{S_2}$ ($S_1 \neq S_2$), and the last step is by (16.6). □

16.6 Reductions between MAXk-LIN(R) problems

Lemma 16.6.1. *Given a q-*BOUNDED-MAXΓ-k-LIN *instance and positive integer $m \geq q$:*

- *When* $k = 2$*,* $\mathsf{opt}_{\mathbb{Z}}(\mathcal{I}), \mathsf{opt}_{\mathbb{R}}(\mathcal{I}), \mathsf{opt}_{\mathbb{Z}_m}(\mathcal{I}) \leq 4 \cdot \mathsf{opt}_{\mathbb{Z}_q}(\mathcal{I})$.

- *When* $k = 3$*,* $\mathsf{opt}_{\mathbb{Z}}(\mathcal{I}), \mathsf{opt}_{\mathbb{R}}(\mathcal{I}), \mathsf{opt}_{\mathbb{Z}_m}(\mathcal{I}) \leq 8 \cdot \mathsf{opt}_{\mathbb{Z}_m}(\mathcal{I})$.

Proof. It is obvious that the $\mathsf{opt}_{\mathbb{Z}}$ is a lower bound for $\mathsf{opt}_{\mathbb{Z}_q}$. It suffice then to show how to convert a δ-good assignment over $\mathbb{Z}_m$ and $\mathbb{R}$ to a $\Omega(\delta)$-good assignment over $\mathbb{Z}$.

First we show the conversion from an assignment over $\mathbb{R}$ to $\mathbb{Z}$. For case of $k = 3$, as is noted in [108], suppose one has an δ-good real assignment to a system of equations of the form $x_{i_1} - x_{i_2} + x_{i_3} = b$, $b \in \mathbb{Z}$. If one randomly rounds each variable up or down to an integer, every formerly satisfied equation has probability at least $1/8$ of remaining satisfied.[1] Hence there must exist a $\delta/8$-good integer assignment. For the case of $k = 2$, The reduction from MAXΓ-2-LIN($\mathbb{Z}$) to MAXΓ-2-LIN($\mathbb{R}$) is even easier and incurs no loss: given a δ-good real assignment, simply dropping the fractional parts yields a δ-good integer assignment.

Next we show the conversion from assignment over $\mathbb{Z}_m$ to $\mathbb{Z}_q$. First let us consider the case of $k = 3$. Suppose one has an δ-good assignment $A : x_i \to Z_m$ to a system of equations of the form

$$x_{i_1} - x_{i_2} + x_{i_3} = b \mod m.$$

Then we know that if $A(x_{i_1}) - A(x_{i_2}) + A(x_{i_3}) = b \mod m$. Notice that $|b| \leq q \leq m$, we must have that $A(x_{i_1}) - A(x_{i_2}) + A(x_{i_3}) \in \{b, b - m, b + m\}$ when the assignment is evaluated over $\mathbb{Z}$. If we define assignments $A_1(x_i) = A(x_i) - m$ and $A_2(x_i) = A(x_i) + m$ for every x_i. Then it is easy to verify that the best assignment among A, A_1, A_2 will give a $\delta/3$-good assignment. Essentially, every equation over Z_m satisfiable by A must also be satisfiable by one of A, A_1, A_2 over $\mathbb{Z}$.

As for the case $k = 2$, we know that for a δ-good assignment A over $\mathbb{Z}_m$, we know that if $A(x_{i_1}) - A(x_{i_2}) = b \mod m$, then $A(x_{i_1}) - A(x_{i_2}) \in \{b - m, b\}$ when evaluated over $\mathbb{Z}$. Therefore, we can randomly set $A'(x_i)$ to be $A(x_i) - m$ or $A(x_i)$. Then we know that A' is at least a $\delta/4$-good assignment over $\mathbb{Z}$.

It is not too hard to see that the proof technique also works for $m < q$; in particular, a δ-good assignment for q-BOUNDED-MAXΓ-k-LINon $\mathbb{Z}_m$ implies a $\Omega(\frac{\delta}{(2q/m)^k})$-good assignment on Z_q. □

[1] In the usual case when the hard instances also have "bipartite" structure, it is not hard to make the loss only a factor of 2 rather than 8.

16.7 From Dictator Tests to hardness of approximation

16.7.1 Proof of Lemma 16.5.2

We start by defining UNIQUEGAMES and the Unique Games Conjecture.

Definition 16.7.1 (UNIQUEGAMES)**.** *A* UNIQUEGAMES *instance $\mathcal{L}(G(U,V,E),\Sigma,\{\pi_e|e\in E\})$ is a constraint satisfaction problem defined as follows. $G(U,V,E)$ is a bipartite graph whose vertices represent variables and edges represent constraints. The goal is to assign to each vertex a label from the set Σ. The constraint on an edge $e=(u,v)\in E$, where $u\in U, v\in V$, is described by a bijection $\pi_e:\Sigma\to\Sigma$. A labeling $\sigma: U\cup V\to\Sigma$ satisfies the constraint on edge $e=(u,v)$ if and only if $\pi_e(\sigma(v))=\sigma(u)$. Let $\mathsf{opt}(\mathcal{U})$ denote the maximum fraction of constraints that can be satisfied by any labeling:*

$$\mathsf{opt}(\mathcal{U}) := \max_{L:U\cup V\to\Sigma}\frac{1}{|E|}\cdot|\{e\in E|L \textit{ satisfies } e\}|.$$

Conjecture 2 (Unique Games Conjecture [136])**.** *For every $\gamma,\delta>0$, there exists a constant $M=M(\gamma,\delta)$, such that given a* UNIQUEGAMES *instance $\mathcal{L}(G(U,V,E),\Sigma,\{\pi_e|e\in E\})$ with $|\Sigma|=M$, it is* NP*-hard to distinguish between these two cases :*

- *YES Case:* $\mathsf{opt}(\mathcal{L})\geq 1-\gamma$.
- *NO Case:* $\mathsf{opt}(\mathcal{L})\leq\delta$.

By standard reductions, we can assume the bipartite graph $G(U,V,E)$ is left-regular in the conjecture.

Now we are ready to prove Lemma 16.5.2.

Proof of Lemma 16.5.2. Given a UNIQUEGAMES instance $\mathcal{L}(G(U,V,E),\Sigma,\{\pi_e|e\in E\})$, and a Dictator Test $\mathcal{T}(\epsilon,\gamma,\kappa,q,K=|\Sigma|)$ described in the lemma statement, we build a q-BOUNDED-MAXΓ-2-LIN instance $\mathcal{I}$ as follows. The variable set consists of all the entries of $g_v:[Q]^\Sigma\to\mathbb{Z},\forall v\in V$, which are supposed Q-ary Long Codes of the labels for $v\in V$, where $Q=q/\gamma$ is defined in the Dictator Test. The equations are placed by the following random process, where the probability of a equation being placed corresponds to its weight.

- Pick a random vertex u and two of its random neighbors of $v,v'\in V$, let $\pi=\pi_{(u,v)}$ and $\pi'=\pi_{(u,v')}$.

- Run the Dictator Test $\mathcal{T}$ on an imaginary function f defined on $[Q]^\Sigma$, suppose $\mathcal{T}$ chooses to test $f(x) - f(y) = b$.
- Place the equation $(g_v \circ \pi)(x) - (g_{v'} \circ \pi')(y) = b$, where $(g \circ \pi)(x) := g(\pi(x))$.

Completeness. Suppose $\mathsf{opt}(\mathcal{L}) \geq 1 - \gamma$, and σ is a labeling function satisfying $1 - \gamma$ fraction of the constraints. Let g_v be the Long Code for $\sigma(v)$, i.e. let $g_v(x) = x_{\sigma(v)}$ for each x. According to the random process shown above, we pick a random equation in $\mathcal{I}$. With probability at least $1-2\gamma$, both of the constraints on (u, v) and (u, v') are satisfied by σ. In this case, both $g_v \circ \pi$ and $g_{v'} \circ \pi'$ are the Long Code for $\sigma(u)$, and $g_v \circ \pi(x) - g_{v'} \circ \pi'(y) = b$ is satisfied with probability $1-\epsilon-\gamma$ by the property of $\mathcal{T}$. In all, at least $1-\epsilon-3\gamma$ fraction (of weight) of the equations are satisfied.

Soundness. Suppose there is a set of functions $g_v : [Q]^\Sigma \to \mathbb{Z}_q$ satisfying more than $q\Lambda_{1-\epsilon}(1/q) + \kappa$ fraction (of weight) of the equations over $\mathbb{Z}_q$. Then there are at least $\kappa/2$ fraction of vertices $u \in U$ such that conditioned on u is picked in the first step of the random process shown above, the equation is satisfied over $\mathbb{Z}_q$ with probability more than $q\Lambda_{1-\epsilon}(1/q) + \kappa/2$. We call such u's "good". For each u, we define $f_u : [Q]^\Sigma \to \Delta_q$ to be $f_u = \text{avg}_{v:(u,v)\in E}\{g_v \circ \pi_{(u,v)}\}$. Since the equations generated after picking u are indeed a Dictator Test $\mathcal{T}$ running on f_u, for good u's, we have $\mathsf{val}_{\mathbb{Z}_q}^{\mathcal{T}}(f_u) > q\Lambda_{1-\epsilon}(1/q) + \kappa/2$. Therefore, for each good u, there exists $i = i_u \in \Sigma$, such that $\mathbf{Inf}_i^{(1-\eta)}[\widetilde{f_u}] > \tau$. Note that

$$\widetilde{f_u} = \underset{v:(u,v)\in E}{\text{avg}} \{\widetilde{g_v \circ \pi_{(u,v)}}\}.$$

By Proposition 16.2.6, we have

$$\tau < \mathbf{Inf}_i^{(1-\eta)}[\widetilde{f_u}] = \mathbf{Inf}_i^{(1-\eta)}\left[\underset{v:(u,v)\in E}{\text{avg}} \{\widetilde{g_v \circ \pi_{(u,v)}}\}\right] \leq \underset{v:(u,v)\in E}{\text{avg}} \left\{\mathbf{Inf}_i^{(1-\eta)}[\widetilde{g_v \circ \pi_{(u,v)}}]\right\}.$$

Therefore, for at $\tau/2$ fraction of neighbors v of u, there exists $j = \pi_{(u,v)}(i)$, such that $\mathbf{Inf}_j^{(1-\eta)}(\widetilde{g_v}) > \tau/2$.

Let $\sigma(u) = i_u$ if u is good. For each $v \in V$, let $\text{Cand}(v) = \{i : \mathbf{Inf}_i^{(1-\eta)}(\widetilde{g_v}) > \tau/2\}$. By Fact 16.2.5, we have $|\text{Cand}(v)| < 1/(\tau\eta)$. If $\text{Cand}(v) \neq \emptyset$, let $\sigma(v)$ be a random element in $\text{Cand}(v)$. Now for a good u, there are $\tau/2$ fraction of neighbors v of u such that $j = \pi_{(u,v)}(\sigma(u)) \in \text{Cand}(v)$, therefore the edge (u, v) is satisfied with probability $1/|\text{Cand}(v)| > \tau\eta$. It follows that σ satisfies more than $(\kappa/2)(\tau/2)\tau\eta = \kappa\eta\tau^2/2$ fraction of the constraints in expectation. Therefore there is a labeling satisfying more than $\delta' = \kappa\eta\tau^2/2$ fraction of the constraints. □

16.7.2 Proof of Lemma 16.5.4

We start by defining Label Cover Games and introducing its hardness.

Definition 16.7.2 (Label Cover Games)**.** *A Label Cover Game* $\mathcal{C}(G(U,V,E),[K],[L],\{\pi_e|e\in E\})$ *is a constraint satisfaction problem defined as follows.* $G(U,V,E)$ *is a bipartite graph whose vertices represent variables and edges represent the constraints. The goal is to assign to each vertex in* U *a label from the set* $[K]$ *and to each vertex in* V *a label from the set* $[L]$*. The constraint on an edge* $e=(u,v)\in E$ *is described by a "projection"* $\pi_e:[L]\to[K]$*. The projection is onto. A labeling* $\sigma:U\to[K],\sigma:V\to[L]$ *satisfies the constraint on edge* $e=(u,v)$ *if and only if* $\pi_e(\sigma(v))=\sigma(u)$*. Let* $\mathsf{opt}(\mathcal{C})$ *denote the maximum fraction of constraints that can be satisfied by any labeling :*

$$\mathsf{opt}(\mathcal{C}) := \max_{\substack{\sigma:U\to[K]\\ \sigma:V\to[L]}} \frac{1}{|E|}\cdot|\{e\in E|\sigma \textit{ satisfies } e\}|.$$

Theorem 16.7.3 (PCP Theorem + Raz's Parallel Repetition Theorem [22, 20, 198])**.** *There exists an absolute constant* c *such that for every* $\delta>0$*, given* $\mathcal{C}(G(U,V,E),[K],[L],\{\pi_e|e\in E\})$*,* $K=(1/\delta)^C$*, it is* NP*-hard to distinguish between:*

- *YES Case:* $\mathsf{opt}(\mathcal{C})=1$.
- *NO Case:* $\mathsf{opt}(\mathcal{C})=\delta$.

Now we are ready to prove Lemma 16.5.4.

Proof of Lemma 16.5.4. Given a Label Cover Game instance $\mathcal{C}(G(U,V,E),[K],[L],\{\pi_e|e\in E\})$, and a Matching Dictator Test $\mathcal{U}(\epsilon,\kappa,q,L,K)$ described in the lemma statement, we build a q-BOUNDED-MAXΓ-3-LIN instance $\mathcal{I}$ as follows. The variable set consists of all the entries of $f_u:[Q]^L\to\mathbb{Z}$ and $g_v:[\mathcal{Q}]^K\to\mathbb{Z}$ for all $u\in U,v\in V$. The equations are the gathering of the Matching Dictator Tests $\mathcal{U}$ for f_u,g_v with projection $\pi_{(u,v)}$ for all $(u,v)\in E$. The weights of the equations are normalised by a factor $1/|E|$.

Completeness. Suppose $\mathsf{opt}(\mathcal{C})=1$, and σ is a labeling function satisfying all the constraints. For all $u\in U,v\in V$, let f_u and g_v be the Long Codes for $\sigma(u),\sigma(v)$ respectively, i.e. let $f_u(x)=x_{\sigma(u)},g_v(y)=y_{\sigma(v)}$. For each edge $(u,v)\in E$, the Matching Dictator Test $\mathcal{U}$ passes with probability at least $1-\epsilon$. Therefore, at least $1-\epsilon$ fraction (of weight) of the equations are satisfied.

Soundness. Suppose there is a set of functions $f_u:[Q]^K\to\mathbb{Z}_q,g_v:[\mathcal{Q}]^L\to\mathbb{Z}_q$ satisfying more than $1/q+\kappa$ fraction (of weight) of the equations over $\mathbb{Z}_q$. By averaging argument, for at least $\kappa/2$ fraction of the edges, the corresponding Matching Dictator Test passes with probability more than $1/q + \kappa/2$. Call these edges "good edges". For all $u \in U, v \in V$, let $\sigma(u) = \mathcal{D}(f_u), \sigma(v) = \mathcal{D}(g_v)$. For good edges $e \in E$, the probability that e is satisfied by σ is at least $\zeta = \zeta(q, \epsilon, \kappa)$. It follows that σ satisfies more than $\zeta\kappa/2$ fraction of the constraints in expectation. Therefore there is a labeling satisfying more than $\delta' = \zeta\kappa/2$ fraction of the constraints.

$\square$

Chapter 17

Hardness of approximating almost satisfiable MAXHORN3-SAT

17.1 Introduction

Schaefer proved long ago that there are only three non-trivial classes of Boolean constraint satisfaction problems (CSPs) for which satisfiability is polynomial time decidable [202]. These are LIN(2) (linear equations modulo 2), 2-SAT, and HORNSAT. The maximization versions of these problems (where the goal is to find an assignment satisfying the maximum number of constraints) are NP-Hard, and in fact APX-Hard, i.e., NP-Hard to approximate within some constant factor bounded away from 1. An interesting special case of the maximization version is the following problem of "finding almost-satisfying assignments": *Given an instance which is $(1-\epsilon)$-satisfiable (i.e., only ϵ fraction of constraints need to be removed to make it satisfiable for some small constant ϵ), can one efficiently find an assignment satisfying most (say, $1 - f(\epsilon) - o(1)$ where $f(\epsilon) \to 0$ as $\epsilon \to 0$) of the constraints?* [1]

The problem of finding almost-satisfying assignments was first suggested and studied in a beautiful paper by Zwick [227]. This problem seems well-motivated, as even if a MAXCSP is APX-Hard in general, in certain practical situations instances might be close to being satisfiable (for example, a small fraction of constraints might have been corrupted

[1]Throughout this chapter, constraints could have weights, and by a "fraction α of constraints" we mean any subset of constraints whose total weight is a fraction α of the sum of the weights of all constraints. For CSPs with no unary constraints, the approximability of the weighted and unweighted versions are known to be the same [71].

by noise). An algorithm that is able to satisfy most of the constraints of such an instance could be very useful.

As pointed out in [135], Schaefer's reductions together with the PCP theorem imply that the previous goal is NP-Hard to achieve for any Boolean CSP for which the satisfiability problem is NP-complete. Indeed, all but the above three tractable cases of Boolean CSPs have a "gap at location 1," which means that given a satisfiable instance it is NP-Hard to find an assignment satisfying α fraction of the constraints for some constant $\alpha < 1$. This result has been extended to CSPs over arbitrary domains recently [128].

The natural question therefore is whether for the three tractable Boolean CSPs, LIN(2), 2-SAT, and HORNSAT, one can find almost-satisfying assignments in polynomial time. Effectively, the question is whether there are "robust" satisfiability checking algorithms that can handle a small number of inconsistent constraints and still produce a near-satisfying assignment.

With respect to the feasibility of finding almost-satisfying assignments, LIN(2), 2-SAT, and HORNSAT behave rather differently from each other. For LIN(2), Håstad in his breakthrough paper [116] showed that for any $\epsilon, \delta > 0$, finding a solution satisfying $1/2 + \delta$ of the equations of a $(1 - \epsilon)$-satisfiable instance is NP-Hard. In fact, this result holds even when each equation depends on only 3 variables. Since just picking a random assignment satisfies $1/2$ the constraints in expectation, this shows, in a very strong sense, that there is no robust satisfiability algorithm for LIN(2).

In sharp contrast to this extreme hardness for linear equations, Zwick [227] proved that for 2-SAT and HORNSAT, one can find almost-satisfying assignments in polynomial time. For MAX2-SAT, Zwick gave a semidefinite programming (SDP) based algorithm that finds a $(1 - O(\epsilon^{1/3}))$-satisfying assignment (i.e., an assignment satisfying a fraction $(1 - O(\epsilon^{1/3}))$ of the constraints) given as input a $(1 - \epsilon)$-satisfiable instance. This algorithm was later improved to one that finds a $1 - O(\sqrt{\epsilon})$-satisfying assignment by Charikar, Makarychev, and Makarychev [62]. The $1 - O(\sqrt{\epsilon})$ bound is known to be best possible under the Unique Games Conjecture (UGC) [136, 141]. In fact, this hardness result for MAX2-SAT was the first application of the UGC and one of the main initial motivations for its formulation by Khot [136].

For HORNSAT, Zwick gave a linear programming (LP) based algorithm to find an assignment satisfying $(1 - O(\log\log(1/\epsilon)/\log(1/\epsilon)))$ of constraints of a $(1 - \epsilon)$-satisfiable instance. Recall that an instance of HORNSAT is a CNF formula where each clause consists of at most one unnegated literal.[2] Equiavlently, each clause is of the form x_i, $\overline{x_i}$, or

[2]The dual variant DUALHORNSAT is an instance of SAT where each clause has at most one negated literal and it is also polynomial time solvable.

$x_i \wedge x_2 \wedge \ldots \wedge x_k \to x_{k+1}$ for variables x_i. For HORN3-SAT where each clause involves at most three variables, the algorithm finds a $(1 - O(1/\log(1/\epsilon)))$-satisfying assignment. Note that the fraction of unsatisfied constraints is *exponentially* worse for HORNSAT compared to MAX2-SAT.

Horn-SAT is a fundamental problem in logic and artificial intelligence. Zwick's robust Horn satisfiability algorithm shows the feasibility of solving instances where a small number of constraints are faulty and raises the following natural question, which was also explicitly raised in [227]. Is this $1/\log(1/\epsilon)$ deficit inherent? Or could a more sophisticated algorithm, say based on an SDP relaxation instead of the LP relaxation used in [227], improve the deficit to something smaller (such as ϵ^b for some constant b as in the case of the SDP based algorithm for MAX2-SAT)? It is known that for some absolute constant $c < 1$, it is NP-Hard to find a $(1 - \epsilon^c)$-satisfying assignment given a $(1 - \epsilon)$-satisfiable instance of HORNSAT [135].

In this chapter, we address the above question and resolve it (conditioned on the UGC), showing the $1/\log(1/\epsilon)$ deficit to be inherent. We describe our results in more detail below in Section 17.2.

Remark 17.1.1. *For $(1 - \epsilon)$-satisfiable instances of* MAX2-SAT*, even the hardness of finding a $(1 - \omega_\epsilon(1)\epsilon)$-satisfying assignment is not known without assuming the UGC (and the UGC implies the optimal $1 - \Omega(\sqrt{\epsilon})$ hardness bound). For* HORNSAT*, as mentioned above, we know the* NP*-Hardness of finding a $(1 - \epsilon^c)$-satisfying assignment for some absolute constant $c < 1$. Under the UGC, we are able to pin down the exact asymptotic dependence on ϵ.*

17.2 Our contributions and previous work

We prove the following hardness result concerning finding almost-satisfying assignments for HORNSAT (in fact for the arity 3 case where all clauses involve at most 3 variables). In the sequel, we use the terminology "UG-Hard" to mean at least as hard as refuting the Unique Games Conjecture.

Theorem 17.2.1. *For some absolute constant $C > 0$, for every $\epsilon > 0$, given a $(1 - \epsilon)$-satisfiable instance of* HORN3-SAT*, it is* UG*-Hard to find an assignment satisfying more than a fraction $\left(1 - \frac{C}{\log(1/\epsilon)}\right)$ of the constraints.*

Zwick gave a polynomial time algorithm that finds a $1 - O(\frac{\log k}{\log(1/\epsilon)})$-satisfying assignment on input a $(1 - \epsilon)$-satisfiable instance of MAXHORNk-SAT. Our inapproximability bound is therefore optimal up to the constant C, and resolves Zwick's question on whether his algorithm can be improved in the negative. (For arbitrary arity HORNSAT, Zwick's algorithm has the slightly worse $1 - O(\log\log(1/\epsilon)/\log(1/\epsilon))$ performance ratio; we do not show this to be tight.)

Theorem 17.2.1 shows that HORNSAT has a very different quantitative behavior compared to MAX2-SAT with respect to approximating near-satisfiable instances: the fraction of unsatisfied clauses $\Omega(1/\log(1/\epsilon))$ is exponentially worse than the $O(\sqrt{\epsilon})$ fraction that can be achieved for MAX2-SAT.

A strong hardness result for MINHORNDELETION, the minimization version for HORNSAT, was shown in [135]. It follows from their reduction that for some absolute constant $c < 1$, it is NP-Hard to find a $(1 - \epsilon^c)$-satisfying assignment given a $(1 - \epsilon)$-satisfiable instance of HORNSAT. The constant c would be extremely close to 1 in this result as it is related to the soundness in Raz's parallel repetition theorem. While our inapproximability bound is stronger and optimal, we are only able to show UG-Hardness and not NP-Hardness.

In light of our strong hardness result for HORN3-SAT, we also consider the approximability of the arity two case. For HORN2-SAT, given a $(1 - \epsilon)$-satisfiable instance, an approximation preserving reduction from vertex cover shows that it is UG-Hard to find a $(1 - c\epsilon)$-satisfying assignment for $c < 2$. It is also shown in [135] that one can find a $(1 - 3\epsilon)$-satisfying assignment efficiently. We improve the algorithmic bound (to the matching UG-Hardness) by proving the following theorem, based on half-integrality of an LP relaxation for the problem.

Theorem 17.2.2. *Given a* $(1 - \epsilon)$*-satisfiable instance for* HORN2-SAT*, it is possible to find a* $(1 - 2\epsilon)$*-satisfying assignment in polynomial time.*

17.3 Proof method

We construct integrality gap instances for a certain semidefinite programming relaxation (described in Section 17.3.1), and then use Raghavendra's theorem [189] to conclude that assuming the Unique Games Conjecture, no algorithm can achieve an approximation ratio better than the SDP integrality gap.

In contrast to previous such integrality gap constructions (eg., for MAXCUT) where the instances had a good SDP solution "by design" and the technical core was bounding the integral optimum, in our case bounding the integral optimum is the easy part and the challenge is in the construction of appropriate SDP vectors. See Section 17.3.2 for an overview of our gap instances. It is also interesting that our SDP gaps match corresponding LP gaps. In general it seems like an intriguing question for which CSPs this is the case and therefore LPs suffice to get the optimal approximation ratio.

For our algorithmic results (see Section 17.3.3), we use a natural linear programming relaxation. The algorithm for HORN2-SAT proceeds by showing half-integrality of the LP.

17.3.1 The canonical SDP for Boolean CSPs and UG-Hardness

For Boolean CSP instances, we write $\mathcal{C}$ as the set of constraints over variables $x_1, x_2, \ldots, x_n \in \{0, 1\}$. The SDP relaxation from [189], which we call the ***canonical SDP***, sets up for each constraint $C \in \mathcal{C}$ a local distribution π_C on all the truth-assignments $\{\sigma : X_C \to \{0, 1\}\}$, where X_C is the set of variables involved in the constraint C. This is implemented via scalar variables $\pi_C(\sigma)$ which are required to be non-negative and satisfy $\sum_{\sigma:X_C\to\{0,1\}} \pi_C(\sigma) = 1$. For each variable x, two orthogonal vectors $\boldsymbol{v}_{(x,0)}$ and $\boldsymbol{v}_{(x,1)}$, corresponding to the events $x = 0$ and $x = 1$, are set up. The SDP requires for each variable x, $\boldsymbol{v}_{(x,0)} \cdot \boldsymbol{v}_{(x,1)} = 0$ and $\boldsymbol{v}_{(x,0)} + \boldsymbol{v}_{(x,1)} = \boldsymbol{I}$ where $\boldsymbol{I}$ is a global unit vector. (In the integral solution, one of the vectors $\boldsymbol{v}_{(x,1)}, \boldsymbol{v}_{(x,0)}$ — based on the x's Boolean value — is intended to be $\boldsymbol{I}$ and the other one to be $\mathbf{0}$.)

Then, as constraint (17.5), the SDP does a consistency check: for two variables x, y (that need not be distinct) involved in the same constraint C, and for every $b_1, b_2 \in \{0, 1\}$, the SDP insists that the inner product $\boldsymbol{v}_{(x,b_1)} \cdot \boldsymbol{v}_{(y,b_2)}$ equals $\Pr_{\sigma\in\pi_C}[(\sigma(x) = b_1) \wedge (\sigma(y) = b_2)]$.

$$\text{Maximize} \qquad \mathop{\mathbf{E}}_{C\in\mathcal{C}}\Big[\Pr_{\sigma\in\pi_C}[C(\sigma) = 1]\Big] \tag{17.1}$$

$$\text{Subject to} \qquad \boldsymbol{v}_{(x_i,0)} \cdot \boldsymbol{v}_{(x_i,1)} = 0 \qquad \forall i \in [n] \tag{17.2}$$

$$\boldsymbol{v}_{(x_i,0)} + \boldsymbol{v}_{(x_i,1)} = \boldsymbol{I} \qquad \forall i \in [n] \tag{17.3}$$

$$\|\boldsymbol{I}\|^2 = 1 \tag{17.4}$$

$$\Pr_{\sigma\in\pi_C}[\sigma(x_i) = b_1 \wedge \sigma(x_j) = b_2] = \boldsymbol{v}_{(x_i,b_1)} \cdot \boldsymbol{v}_{(x_j,b_2)} \quad \forall C \in \mathcal{C}, x_i, x_j \in C, \; b_1, b_2 \in \{0, 1\} \tag{17.5}$$

Note that if we discard all the vectors by removing constraints (17.2)∼(17.4), and changing constraints (17.5) to $\Pr_{\sigma\in\pi_S}[\sigma(x_i) = b_1 \wedge \sigma(x_j) = b_2] = X_{(x_i,b_1),(x_j,b_2)}$, the SDP becomes a lifted LP in Sherali-Adams system. We call this LP scheme the *lifted LP* in this chapter.

The following striking theorem (Theorem 1.1 in [189]) states that once we have an integrality gap for the canonical SDP, we also get a matching UG-Hardness. Below and elsewhere in the chapter, a c vs. s gap instance is an instance with SDP optimum at least c and integral optimum at most s.

Theorem 17.3.1. *Let $1 > c > s > 0$. If a constraint satisfaction problem Λ admits a c vs. s integrality gap instance for the above canonical SDP, then for every constant $\eta > 0$, given an instance of Λ that admits an assignment satisfying $(c - \eta)$ of constraints, it is* UG*-Hard to find an assignment satisfying more than $(s + \eta)$ of constraints.*

To make our construction of integrality gaps easier, we notice the following simplification of the above SDP. Suppose we are given the global unit vector $\boldsymbol{I}$ and a vector $\boldsymbol{v}_x$ for each variable x in the CSP instance, subject to the following constraints:

$$(\boldsymbol{I} - \boldsymbol{v}_x) \cdot \boldsymbol{v}_x = 0 \qquad \forall \text{ variables } x \tag{17.6}$$

$$\Pr_{\sigma \in \pi_C}[\sigma(x_i) = 1 \wedge \sigma(x_j) = 1] = \boldsymbol{v}_{x_i} \cdot \boldsymbol{v}_{x_j} \qquad \forall C \in \mathcal{C}, x_i, x_j \in C\,. \tag{17.7}$$

Defining $\boldsymbol{v}_{(x,1)} = \boldsymbol{v}_x$ and $\boldsymbol{v}_{(x,0)} = \boldsymbol{I} - \boldsymbol{v}_x$, it is easy to check that all constraints of the above SDP are satisfied. For instance, for variables x, y belonging to a constraint C,

$$\begin{aligned}\boldsymbol{v}_{(x,0)} \cdot \boldsymbol{v}_{(y,1)} &= (\boldsymbol{I} - \boldsymbol{v}_{(x,1)}) \cdot \boldsymbol{v}_{(y,1)} \\ &= \|\boldsymbol{v}_{(y,1)}\|^2 - \boldsymbol{v}_{(x,1)} \cdot \boldsymbol{v}_{(y,1)} \\ &= \Pr_{\sigma \in \pi_C}[\sigma(y) = 1] - \Pr_{\sigma \in \pi_C}[(\sigma(x) = 1) \wedge (\sigma(y) = 1)] \\ &= \Pr_{\sigma \in \pi_C}[(\sigma(x) = 0) \wedge (\sigma(y) = 1)]\,,\end{aligned}$$

and other constraints of (17.5) follow similarly.

Henceforth in this chapter, we will work with this streamlined canonical SDP with vector variables $\boldsymbol{I}$, $\{\boldsymbol{v}_x\}$, scalar variables corresponding to the local distributions π_C, constraints (17.6) and (17.7), and objective function (17.1).

17.3.2 Overview of construction of SDP gaps

In the concluding section of [227], Zwick remarks that there is an integrality gap for the LP he uses that matches his approximation ratio. Indeed such a LP gap is not hard to construct and we start by describing one such instance. The instance begins with clause x_1, and in the intermediate $(k-1)$ clauses, the i-th clause $x_1 \wedge \ldots \wedge x_i \rightarrow x_{i+1}$ makes x_{i+1} true if all the previous clauses are satisfied. Then the last clause $\overline{x_k}$ generates a contradiction. Thus the optimal integral solution is at most $(1 - 1/k)$. On the other hand, one possible fractional solution starts with $x_1 = (1 - \epsilon)$ for some $\epsilon > 0$. Then for $1 \leq i < k$, by letting

$(1-x_{i+1}) = \sum_{j=1}^{i}(1-x_j)$, all the intermediate $(k-1)$ clauses are perfectly "satisfied" by the LP, while the gap $(1-x_{i+1}) = 2^{i-1}\epsilon$ increases exponentially. Thus by letting $\epsilon = 1/2^{k-2}$, we get $x_k = 0$ and the LP solution is at least $(1-1/2^{\Omega(k)})$. The instance gives a $(1-2^{-\Omega(k)})$ vs. $(1-1/k)$ LP integrality gap.

Now we convert this LP gap instance into an SDP gap instance in two steps. First, since we are going to give a gap instance for MAX HORN-3SAT, we reduce the arity of the instance from k to 3. Then, we find a set of vectors for the LP solution to make it an SDP solution.

For the first step, to get an instance of MAX HORN-3SAT, we introduce y_i which is intended to be $x_1 \wedge \ldots \wedge x_{i-1}$. For $1 \leq i < k$, we replace the intermediate clauses by $x_i \wedge y_i \rightarrow x_{i+1}$, and add $x_i \wedge y_i \rightarrow y_{i+1}$ to meet the intended definition of y_i. We call each of these two clauses as comprising one step . It is easy to show that for this instance there is a solution of value $(1-1/2^{\Omega(k)})$ even for the lifted LP. (The difference between the lifted LP and Zwick's LP is that the lift LP introduces local distributions over clauses which are consistent in the first and second moments, while Zwick's LP only has variables for singletons.)

Finding vectors for the SDP turns out to be more challenging. Note that if we want to perfectly satisfy all the intermediate clauses in SDP, we need to obey $\boldsymbol{v}_{x_i} \cdot \boldsymbol{v}_{y_i} \leq \|\boldsymbol{v}_{x_{i+1}}\|^2$ and $\boldsymbol{v}_{x_i} \cdot \boldsymbol{v}_{y_i} \leq \|\boldsymbol{v}_{y_{i+1}}\|^2$ for $1 \leq i < k$. Thus to make the norms $\|\boldsymbol{v}_{x_{i+1}}\|^2$ and $\|\boldsymbol{v}_{y_{i+1}}\|^2$ decrease fast (since we want $\|\boldsymbol{v}_{x_k}\|^2 = \|\boldsymbol{v}_{y_k}\|^2 = 0$), we need to make the inner product $\boldsymbol{v}_{x_i} \cdot \boldsymbol{v}_{y_i}$ decrease fast as well. But technically it is hard to make both kinds of quantities decrease at a high rate for all intermediate clauses. Our solution is to decrease the norms and inner products alternately. More specifically, we divide the intermediate clauses into blocks, each of which contains two consecutive steps. In the first step of each block, we need that the inner product is much smaller than the norms so that we can decrease the norms quickly, but we preserve the value of inner product. Thus we cannot do this step repeatedly, and we need the second step, where we decrease the inner product (while preserving the norms) in preparation to start the first step of the next block.

17.3.3 Overview of algorithmic results

Our algorithmic result for HORN2-SAT (Theorem 17.2.2) is obtained by rounding fractional solutions of appropriate linear programming (LP) relaxations. The algorithm is indeed a 2-approximation algorithm for MINHORNDELETION problem (refer to Section 17.4.2 for the definition of MINHORNDELETION). We prove a half-integrality property of the optimal solution to the natural LP relaxation of the problem, which can be viewed as a generalization of half-integrality property of (the natural LP for) Vertex Cover. We take the optimal solution of the natural LP relaxation, iteratively make every variable move towards half-integral values (0, 1, and $1/2$), while never increasing the value of the solution. This yields an optimal half-integral solution which can then be trivially rounded to obtain an integral solution that gives a factor 2 approximation.

17.4 Approximability of HORN3-SAT

17.4.1 SDP gap and UG hardness for HORN3-SAT

17.4.1.1 Instance

We consider the following HORN3-SAT instance $\mathcal{I}_k^{\mathrm{Horn}}$ parameterized by $k \geq 1$. (This construction is essentially the same as the one described in Section 17.3.2.)

$$\begin{array}{rrrr} \text{Start point:} & & & x_0, y_0 \\ \text{Block } i\ (0 \leq i \leq k-1) & \text{Step } i.1: & x_{2i} \wedge y_{2i} \to x_{2i+1}, & x_{2i} \wedge y_{2i} \to y_{2i+1} \\ & \text{Step } i.2: & x_{2i+1} \wedge y_{2i+1} \to x_{2i+2}, & x_{2i+1} \wedge y_{2i+1} \to y_{2i+2} \\ \text{End point:} & & x_{2k} \wedge y_{2k} \to x_{2k+1}, & x_{2k} \wedge y_{2k} \to y_{2k+1} \\ & & \overline{x_{2k+1}}, & \overline{y_{2k+1}} \end{array}$$

It is easy to see this instance contains $(4k+6)$ clauses, and cannot be completely satisfied. Thus we have:

Lemma 17.4.1. *Every Boolean assignment satisfies at most a fraction* $1 - 1/(4k+6)$ *of the clauses of* $\mathcal{I}_k^{\mathrm{Horn}}$.

17.4.1.2 Construction of a good SDP solution

We will work with the SDP in simplified form described at the end of Section 17.3.1. Recall that the SDP requires a local distribution for each clause, and uses vectors to check the consistency on every pair of variables that belong to the clause. To construct a good solution for the SDP, we want to first find a good solution in the scalar part (i.e., local distributions), and then construct vectors which meet the consistency requirement. But it is difficult to construct a lot of vectors which meet all the requirements simultaneously. Thus, we break down the whole construction task into small pieces, each of which is easy to deal with. As long as there are solutions to these small pieces, and the solutions agree with each other on some interfaces, we can coalesce the small solutions together and come up with a global solution. The following definition and claim formally help us bring down the difficulty, and focus on one local block of variables at a time.

Definition 17.4.2 (partial solution)**.** *Let $\mathcal{C}' \subseteq \mathcal{C}$ be a subset of clauses. $f = \{\pi_C = \pi_C(f), \boldsymbol{v}_x = \boldsymbol{v}_x(f), \boldsymbol{I} = \boldsymbol{I}(f) \mid \forall C \in \mathcal{C}', x \in C\}$ is said to be a* partial solution *on $\mathcal{C}'$, if all constraints of the SDP restricted to the subset of variables defined in f are satisfied.*

Claim 17.4.3. *Let $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{C}$ be two disjoint set of clauses. Let f and g be partial solutions on $\mathcal{C}_1, \mathcal{C}_2$ respectively. If for all $\boldsymbol{v}_1, \boldsymbol{v}_2$ (not necessarily distinct) defined in both f and g, $\boldsymbol{v}_1(f) \cdot \boldsymbol{v}_2(f) = \boldsymbol{v}_1(g) \cdot \boldsymbol{v}_2(g)$, then there exists a partial solution, namely h, for $\mathcal{C}_1 \cup \mathcal{C}_2$, such that $\forall C_1 \in \mathcal{C}_1, C_2 \in \mathcal{C}_2$, $\pi_{C_1}(h) = \pi_{C_1}(f), \pi_{C_2}(h) = \pi_{C_2}(g)$.*

Proof. Let X be the set of variables x for which $\boldsymbol{v}_x(f)$ and $\boldsymbol{v}_x(g)$ are both defined. Denote $V_f = \{\boldsymbol{v}_x(f) \mid x \in X\} \cup \{\boldsymbol{I}(f)\}$ and $V_g = \{\boldsymbol{v}_x(g) \mid x \in X\} \cup \{\boldsymbol{I}(g)\}$. Since the dot products of every pair of vectors in V_f exactly equals the dot product between the corresponding pair in V_g, there is a rotation (orthogonal transformation) T such that $\boldsymbol{I}(f) = T\boldsymbol{I}(g)$ and for all $x \in X$, $\boldsymbol{v}_x(f) = T\boldsymbol{v}_x(g)$.

Now define the partial solution g' as $\pi_C(g') = \pi_C(g)$ for all $C \in \mathcal{C}_2$ and $\boldsymbol{v}_x(g') = T\boldsymbol{v}_x(g), \boldsymbol{I}(g') = T\boldsymbol{I}(g)$ for all $x \in C \in \mathcal{C}_2$. Obviously f and g' agree on all the scalar and vector variables that are defined in both f and g'. Letting

$$\boldsymbol{v}_x(h) = \begin{cases} \boldsymbol{v}_x(f) & x \in C \in \mathcal{C}_1 \\ \boldsymbol{v}_x(g') & x \in C \in \mathcal{C}_2 \end{cases}, \pi_C(h) = \begin{cases} \pi_C(f) & C \in \mathcal{C}_1 \\ \pi_C(g') & C \in \mathcal{C}_2 \end{cases},$$

it is easy to see h is a partial solution on $\mathcal{C}_1 \cup \mathcal{C}_2$. □

By the above lemma, if we establish the following lemma which constructs a good partial solution on each block (the proof of which is deferred to Section 17.4.1.3), it is then easy to get a good global solution.

Lemma 17.4.4. *For each Block i $(0 \le i \le k-1)$, each $0 < c \le 0.2$, let $r_c = 1.5(1+c)/(1.5+c) > 1$, and for each $0 < p \le \frac{1}{(1+c)r_c}$, there is a partial solution f which completely satisfies all the clauses in Block i (by local distributions), and with following properties,*

$$\begin{aligned} \|\boldsymbol{v}_{x_{2i}}(f)\|^2 = \|\boldsymbol{v}_{y_{2i}}(f)\|^2 =& 1 - p \\ \boldsymbol{v}_{x_{2i}}(f) \cdot \boldsymbol{v}_{y_{2i}}(f) =& 1 - (1+c)p \\ \|\boldsymbol{v}_{x_{2i+2}}(f)\|^2 = \|\boldsymbol{v}_{y_{2i+2}}(f)\|^2 =& 1 - r_c p \\ \boldsymbol{v}_{x_{2i+2}}(f) \cdot \boldsymbol{v}_{y_{2i+2}}(f) =& 1 - (1+c)r_c p. \end{aligned}$$

As explained in Section 17.3.2, in the first step (the step to decrease norms), to make $\|\boldsymbol{v}_{x_{2i+2}}(f)\|^2$ and $\|\boldsymbol{v}_{y_{2i+2}}(f)\|^2$ much smaller than $\|\boldsymbol{v}_{x_{2i}}(f)\|^2$ and $\|\boldsymbol{v}_{y_{2i}}(f)\|^2$, we need the inner product $\boldsymbol{v}_{x_{2i}}(f) \cdot \boldsymbol{v}_{y_{2i}}(f)$ to be small. This is why we introduce c, and require that $\boldsymbol{v}_{x_{2i}}(f) \cdot \boldsymbol{v}_{y_{2i}}(f) = 1 - (1+c)p$. The larger c is, the faster the norms decrease. But due to technical reasons, in the second step (the step to decrease the inner product), we are not able to decrease the inner product fast when it is much smaller than the norms. So we put a upper bound $c \leq 0.2$ in the lemma.

Using Lemma 17.4.4 together with Claim 17.4.3, we immediately get the following corollary.

Corollary 17.4.5. *For the union of Block* 0 *to Block* k' $(0 \leq k' \leq k-1)$, *given parameters* $0 < c \leq 0.2$ *and* $0 < p \leq \frac{1}{(1+c)r_c^{k'+1}}$, *there is a partial solution* g *which completely satisfies all the clauses, and with following properties,*

$$\begin{aligned}
\|\boldsymbol{v}_{x_0}(g)\|^2 = \|\boldsymbol{v}_{y_0}(g)\|^2 =& 1 - p \\
\boldsymbol{v}_{x_0}(g) \cdot \boldsymbol{v}_{y_0}(g) =& 1 - (1+c)p \\
\|\boldsymbol{v}_{x_{2k'+2}}(g)\|^2 = \|\boldsymbol{v}_{y_{2k'+2}}(g)\|^2 =& 1 - r_c^{k'+1}p \\
\boldsymbol{v}_{x_{2k'+2}}(g) \cdot \boldsymbol{v}_{y_{2k'+2}}(g) =& 1 - (1+c)r_c^{k'+1}p.
\end{aligned}$$

Proof. Apply induction on k'. The basis case $k' = 0$ is exactly Lemma 17.4.4. For $k' > 0$, by induction hypothesis there is a partial solution g' satisfying all the clauses of the union of Blocks 0 to $k'-1$ with the same parameter c, p. By Lemma 17.4.4, there is a partial solution f satisfying all the clauses of Block k' with parameter $c, r_c^{k'}p$. Since g' and f agree on pairwise inner-products over the definition of $\{\boldsymbol{v}_{x_{2k'}}, \boldsymbol{v}_{y_{2k'}}\}$, by Claim 17.4.3, there is a partial solution g on the union of Blocks 0 to k' completely satisfying all the clauses. □

With the above pieces in place, we now come to the final SDP solution.

Lemma 17.4.6. *The optimal SDP solution for the instance* $\mathcal{I}_k^{\mathrm{Horn}}$ *has value at least* $1 - \frac{1}{(2k+3)1.05^k}$.

Proof. By Corollary 17.4.5, for any $0 < c \leq 0.2$, by setting $p = \frac{1}{(1+c)r_c^k}$, there is a partial solution g completely satisfying all the clauses of all the blocks, with

$$\begin{aligned}
\|\boldsymbol{v}_{x_0}(g)\|^2 = \|\boldsymbol{v}_{y_0}(g)\|^2 =& 1 - \frac{1}{(1+c)r_c^k} \\
\|\boldsymbol{v}_{x_{2k}}(g)\|^2 = \|\boldsymbol{v}_{y_{2k}}(g)\|^2 =& c/(1+c) \\
\boldsymbol{v}_{x_{2k}}(g) \cdot \boldsymbol{v}_{y_{2k}}(g) =& 0.
\end{aligned}$$

Based on g, we define a local distribution on two "Start point" clauses by making x_0 (or y_0) equal 1 with probability $1-p$. At "End point", we define the local distribution on clause $x_{2k} \wedge y_{2k} \to x_{2k+1}$ as

$$\Pr_{\pi}[x_{2k} = 1 \wedge y_{2k} = 0 \wedge x_{2k+1} = 0] = c/(1+c)$$
$$\Pr_{\pi}[x_{2k} = 0 \wedge y_{2k} = 1 \wedge x_{2k+1} = 0] = c/(1+c)$$
$$\Pr_{\pi}[x_{2k} = 0 \wedge y_{2k} = 0 \wedge x_{2k+1} = 0] = (1-c)/(1+c)\ .$$

And a similar distribution for the clause $x_{2k} \wedge y_{2k} \to y_{2k+1}$ can be defined (by replacing x_{2k+1} by y_{2k+1} in the equations above). The distribution on clauses $\overline{x_{2k+1}}$ and $\overline{y_{2k+1}}$ never picks the corresponding variable to be 1. By defining $\boldsymbol{v}_{x_{2k+1}}$ and $\boldsymbol{v}_{y_{2k+1}}$ to be zero vectors, we note that the distributions are consistent with vectors. Thus the solution we construct is valid.

On the other hand, note that all the distributions locally satisfy the clauses, except for the distributions at "Start point" satisfy the corresponding clause with probability $1 - \frac{1}{(1+c)r_c^k}$. Thus the SDP solution has value $1 - \frac{2}{(4k+6)(1+c)r_c^k} = 1 \geq 1 - \frac{1}{(2k+3)r_c^k}$. By setting $c = 0.2$, we get $r_c \geq 1.05$. Thus the best SDP solution has value better than $1 - \frac{1}{(2k+3)1.05^k}$. $\square$

Combining Lemma 17.4.1 and Lemma 17.4.6, we get the following theorem.

Theorem 17.4.7. *$\mathcal{I}_k^{\mathrm{Horn}}$ is a $(1-\epsilon)$ vs. $(1-\Omega(1/\log(1/\epsilon)))$ gap instance of* HORN3-SAT *for the canonical SDP relaxation.*

Together with Theorem 17.3.1, Theorem 17.4.7 implies our main result, Theorem 17.2.1, on HORNSAT.

17.4.1.3 Proof of the Key Lemma 17.4.4

For Block i, denote the clauses in Step $i.1$ by C_{1x} and C_{1y}, and the clauses in Step $i.2$ by C_{2x} and C_{2y}. We first construct partial solutions on Step $i.1$ and Step $i.2$ separately, as follows.

Partial solution on Step i.1 We first define a local distribution on satisfying assignments for C_{1x} as follows, and C_{1y} in a similar way (by replacing x_{2i+1} by y_{2i+1} in following equations).

$$\Pr_{\pi_{C_{1x}}}[x_{2i} = 1 \wedge y_{2i} = 1 \wedge x_{2i+1} = 1] = 1-(1+c)p$$

$$\Pr_{\pi_{C_{1x}}}[x_{2i}=1 \wedge y_{2i}=0 \wedge x_{2i+1}=0] = cp$$
$$\Pr_{\pi_{C_{1x}}}[x_{2i}=0 \wedge y_{2i}=1 \wedge x_{2i+1}=0] = cp$$
$$\Pr_{\pi_{C_{1x}}}[x_{2i}=0 \wedge y_{2i}=0 \wedge x_{2i+1}=1] = (1+c-r_c)p = \frac{(1+c)c}{1.5+c}\cdot p$$
$$\Pr_{\pi_{C_{1x}}}[x_{2i}=0 \wedge y_{2i}=0 \wedge x_{2i+1}=0] = (r_c-2c)p = \frac{1.5-1.5c-2c^2}{1.5+c}\cdot p.$$

Recall $r_c = 1.5(1+c)/(1.5+c)$. Note that all the probabilities are defined to be non-negative values by the range of c and p, and they sum up to 1.

We observe the following inner-product matrix A over $\boldsymbol{I}, \boldsymbol{v}_{x_{2i}}, \boldsymbol{v}_{y_{2i}}, \boldsymbol{v}_{x_{2i+1}}, \boldsymbol{v}_{y_{2i+1}}$ is consistent with the local distributions on satisfying assignments for C_{1x} and C_{1y}.

$$A = \begin{bmatrix} 1 & 1-p & 1-p & 1-r_cp & 1-r_cp \\ 1-p & 1-p & 1-(1+c)p & 1-(1+c)p & 1-(1+c)p \\ 1-p & 1-(1+c)p & 1-p & 1-(1+c)p & 1-(1+c)p \\ 1-r_cp & 1-(1+c)p & 1-(1+c)p & 1-r_cp & 1-(1+c)p \\ 1-r_cp & 1-(1+c)p & 1-(1+c)p & 1-(1+c)p & 1-r_cp \end{bmatrix}$$

By Claim 17.4.8 (at the end of this section) we know that A is positive semidefinite, and therefore there is a set of vectors consistent with our local distributions, i.e., we get a partial solution on Step i.1.

Partial solution on Step i.2 We define the local distribution on satisfying assignments for C_{2x} as follows. The distribution for C_{2y} is defined in a similar way (by replacing x_{2i+2} with y_{2i+2} in the following equations). Let $q = r_cp$ and $\epsilon = c/1.5$.

$$\begin{aligned}
\Pr_{\pi_{C_{2x}}}[x_{2i+1}=1 \wedge y_{2i+1}=1 \wedge x_{2i+2}=1] &= 1-(1+\epsilon)q \\
\Pr_{\pi_{C_{2x}}}[x_{2i+1}=1 \wedge y_{2i+1}=0 \wedge x_{2i+2}=0] &= \epsilon q \\
\Pr_{\pi_{C_{2x}}}[x_{2i+1}=0 \wedge y_{2i+1}=1 \wedge x_{2i+2}=0] &= \epsilon q \\
\Pr_{\pi_{C_{2x}}}[x_{2i+1}=0 \wedge y_{2i+1}=0 \wedge x_{2i+2}=1] &= \epsilon q \\
\Pr_{\pi_{C_{2x}}}[x_{2i+1}=0 \wedge y_{2i+1}=0 \wedge x_{2i+2}=0] &= (1-2\epsilon)q.
\end{aligned}$$

Note that all the probabilities are defined to be non-negative values by the range of c and p, and they sum up to 1.

Then note that the following inner-product matrix B over $\boldsymbol{I}, \boldsymbol{v}_{x_{2i+1}}, \boldsymbol{v}_{y_{2i+1}}, \boldsymbol{v}_{x_{2i+2}}, \boldsymbol{v}_{y_{2i+2}}$ is consistent with the local distribution.

$$B = \begin{bmatrix} 1 & 1-q & 1-q & 1-q & 1-q \\ 1-q & 1-q & 1-(1+\epsilon)q & 1-(1+\epsilon)q & 1-(1+\epsilon)q \\ 1-q & 1-(1+\epsilon)q & 1-q & 1-(1+\epsilon)q & 1-(1+\epsilon)q \\ 1-q & 1-(1+\epsilon)q & 1-(1+\epsilon)q & 1-q & 1-(1+1.5\epsilon)q \\ 1-q & 1-(1+\epsilon)q & 1-(1+\epsilon)q & 1-(1+1.5\epsilon)p & 1-q \end{bmatrix}$$

Again by Claim 17.4.8, B is positive semidefinite, and therefore there is a set of vectors consistent with local distributions – we have constructed a partial solution on Step $i.2$.

Combining the two partial solutions. We first check that under our parameter setting, partial solutions on Step $i.1$ and Step $i.2$ agree on pairwise inner-products between their shared vectors $\boldsymbol{I}, \boldsymbol{v}_{x_{2i+1}}, \boldsymbol{v}_{y_{2i+1}}$.

- For $\langle \boldsymbol{I}, \boldsymbol{I} \rangle$, we have $1 = 1$.
- For $\langle \boldsymbol{I}, \boldsymbol{v}_{x_{2i+1}} \rangle$, we have $1 - r_c p = 1 - q$.
- For $\langle \boldsymbol{I}, \boldsymbol{v}_{y_{2i+1}} \rangle$, we have $1 - r_c p = 1 - q$.
- For $\langle \boldsymbol{v}_{x_{2i+1}}, \boldsymbol{v}_{x_{2i+1}} \rangle$, we have $1 - r_c p = 1 - q$.
- For $\langle \boldsymbol{v}_{x_{2i+1}}, \boldsymbol{v}_{y_{2i+1}} \rangle$, we have $1 - (1+c)p = 1 - (1 + c/1.5) r_c p = 1 - (1+\epsilon)q$.
- For $\langle \boldsymbol{v}_{y_{2i+1}}, \boldsymbol{v}_{y_{2i+1}} \rangle$, we have $1 - r_c p = 1 - q$.

Thus, there is a partial solution on Block i, with

$$\begin{aligned} \|\boldsymbol{v}_{x_{2i}}(f)\|^2 = \|\boldsymbol{v}_{y_{2i}}(f)\|^2 =& 1 - p \\ \boldsymbol{v}_{x_{2i}}(f) \cdot \boldsymbol{v}_{y_{2i}}(f) =& 1 - (1+c)p \\ \|\boldsymbol{v}_{x_{2i+2}}(f)\|^2 = \|\boldsymbol{v}_{y_{2i+2}}(f)\|^2 =& 1 - q = 1 - r_c p \\ \boldsymbol{v}_{x_{2i+2}}(f) \cdot \boldsymbol{v}_{y_{2i+2}}(f) =& 1 - (1+1.5\epsilon)q = 1 - (1+c) r_c p. \quad \square \end{aligned}$$

Finally, we establish the two positive semidefinite matrices used in the proof above.

Claim 17.4.8. *Given* $0 < c \leq 0.2, 0 < p \leq \frac{1}{1+c} r_c, q = r_c p, \epsilon = c/1.5$*, the following two matrices are positive semidefinite.*

$$A = \begin{bmatrix} 1 & 1-p & 1-p & 1-r_c p & 1-r_c p \\ 1-p & 1-p & 1-(1+c)p & 1-(1+c)p & 1-(1+c)p \\ 1-p & 1-(1+c)p & 1-p & 1-(1+c)p & 1-(1+c)p \\ 1-r_c p & 1-(1+c)p & 1-(1+c)p & 1-r_c p & 1-(1+c)p \\ 1-r_c p & 1-(1+c)p & 1-(1+c)p & 1-(1+c)p & 1-r_c p \end{bmatrix},$$

$$B=\begin{bmatrix} 1 & 1-q & 1-q & 1-q & 1-q \\ 1-q & 1-q & 1-(1+\epsilon)q & 1-(1+\epsilon)q & 1-(1+\epsilon)q \\ 1-q & 1-(1+\epsilon)q & 1-q & 1-(1+\epsilon)q & 1-(1+\epsilon)q \\ 1-q & 1-(1+\epsilon)q & 1-(1+\epsilon)q & 1-q & 1-(1+1.5\epsilon)q \\ 1-q & 1-(1+\epsilon)q & 1-(1+\epsilon)q & 1-(1+1.5\epsilon)p & 1-q \end{bmatrix}.$$

Proof. Let J be the all 1 matrix, E_1 be the matrix with 1 in entry $(1,1)$ as the only one non-zero entry. We also define $E_{i,j}, F_{i,j}$ and $G_{i,j}$ as matrices with only four non-zero entries located in the intersections of Column i,j and Row i,j. The sub-matrices of $E_{i,j}, F_{i,j}$ and $G_{i,j}$ on Column i,j and Row i,j are defined as

$$(\text{for } E_{i,j}) \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, (\text{for } F_{i,j}) \begin{bmatrix} 2 & 1 \\ 1 & 0.5 \end{bmatrix} \text{ and } (\text{for } G_{i,j}) \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Clearly, all of J, E_1, $E_{i,j}$, $F_{i,j}$ and $G_{i,j}$ are positive semidefinite matrices.

Then we can write A as

$$\begin{aligned} A =& (1-(1+c)p)J + cp(E_{1,2}+E_{1,3}) + (1+c-r_c)p(E_{1,4}+E_{1,5}) + (2r_c-1-3c)pE_1 \\ =& (1-(1+c)p)J + cp(E_{1,2}+E_{1,3}) + \frac{(1+c)c}{1.5+c}\cdot p(E_{1,4}+E_{1,5}) + \frac{1.5-2.5c-3c^2}{1.5+c}\cdot pE_1, \end{aligned}$$

Note that all the coefficient before matrices are non-negative within the range of c. Since A can be written as the sum of several positive semidefinite matrices, A is positive semidefinite.

For matrix B, note that

$$B = (1-(1+\epsilon)q)J + \epsilon q(E_{1,2}+E_{1,3}+F_{1,4}+F_{1,5}) + 0.5\epsilon q G_{4,5} + (1-5\epsilon)qE_1,$$

Clearly, as long as $5\epsilon = 5c/1.5 < 1$, B can be expressed as sum of positive semidefinite matrices, and hence B is positive semidefinite. □

17.4.2 Algorithm for MINHORNDELETION and MAXHORN2-SAT

In the MINHORNDELETION problem, we are given a HORN2-SAT instance, and the goal is to find a subset of clauses of minimum total weight whose deletion makes the instance satisfiable. A factor 3 approximation algorithm for MINHORNDELETION is given in [135]. Here we improve the approximation ratio to 2. Note that following reduction from vertex cover is approximation preserving: given a graph with n vertices, for each edge (i, j), introduce a clause $\bar{x_i} \vee \bar{x_j}$ of weight n; for each vertex i, introduce a clause x_i of weight 1. Therefore, by the inapproximability result for vertex cover [143], our approximation algorithm for MINHORNDELETION is optimal under the UGC.

Our motivation to study MINHORNDELETION in the context of this chapter is to pin down the fraction of clauses one can satisfy in a $(1-\epsilon)$-satisfiable instance of HORN2-SAT: we can satisfy a fraction $(1-2\epsilon)$ of clauses (even in the weighted case), and satisfying a $(1-c\epsilon)$ fraction is hard for $c < 2$ assuming that vertex cover does not admit a c-approximation for any constant $c < 2$.

In this section, we prove the following theorem by showing half-integrality of a natural LP relaxation for the problem.

Theorem 17.4.9. *There is a polynomial-time* 2*-approximation algorithm for* MINHORNDELETION *problem.*

A direct corollary of Theorem 17.4.9 is the following result for approximating near-satisfiable instances of HORN2-SAT.

Theorem 17.2.2 (restated). *Given a* $(1-\epsilon)$*-satisfiable instance for* MAXHORN2-SAT*, it is possible to find a* $(1-2\epsilon)$*-satisfying assignment efficiently.*

17.4.2.1 LP Formulation

We find it slightly more convenient to present the algorithm for DUALHORN2-SAT where each clause has at most one negated literal. (So the clauses are of the form x, $\bar{x}$, $x \vee y$, or $x \to y$, for variables x, y.) Let $w_{ij}^{(D)} > 0$ be the weight imposed on the disjunction constraint $x_i \vee x_j$ (for each pair of i, j such that $i < j$), and $w_{ij}^{(I)} > 0$ be the weight imposed on the implication constraint $x_i \to x_j$ (for each pair of i, j such that $i \neq j$). For each variable x_i, let $w_i^{(T)}$ be the weight on x_i being true (i.e. $x_i = 1$), and $w_i^{(F)}$ be the weight on x_i being false (i.e. $x_i = 0$). Then we write the following LP relaxation, where each real variable y_i corresponds to the integer variable x_i.

$$
\begin{array}{lll}
\text{Minimize} & \multicolumn{2}{l}{\sum_{i\in V} w_i^{(T)}(1-y_i) + \sum_{i\in V} w_i^{(F)} y_i + \sum_{i<j} w_{ij}^{(D)} z_{ij}^{(D)} + \sum_{i\neq j} w_{ij}^{(I)} z_{ij}^{(I)}} \\
\text{Subject to} & z_{ij}^{(D)} \geq 1 - y_i - y_j & \forall i < j \\
 & z_{ij}^{(I)} \geq y_i - y_j & \forall i \neq j \\
 & z_{ij}^{(D)} \geq 0 & \forall i < j \\
 & z_{ij}^{(I)} \geq 0 & \forall i \neq j \\
 & y_i \in [0,1] & \forall i \in V
\end{array}
$$

Let OPT be the optimal value of the integral solution, and $\mathrm{OPT}_{\mathrm{LP}}$ be the optimal value of the LP solution. We have $\mathrm{OPT}_{\mathrm{LP}} \leq \mathrm{OPT}$.

17.4.2.2 Half-integrality and rounding

Given a LP solution $f = \{z_{ij}^{(D)}, z_{ij}^{(I)}, y_i\}$, we can assume $z_{ij}^{(D)} = \max\{1 - y_i - y_j, 0\}$ and $z_{ij}^{(I)} = \max\{y_i - y_j, 0\}$ to minimize $\mathsf{val}(f)$. Thus, we only need $f = \{y_i\}$ to characterize a solution, and we have

$$
\mathsf{val}(f) = \\
\sum_{i\in V} w_i^{(T)}(1-y_i) + \sum_{i\in V} w_i^{(F)} y_i + \sum_{i<j} w_{ij}^{(D)} \max\{1-y_i-y_j, 0\} + \sum_{i\neq j} w_{ij}^{(I)} \max\{y_i - y_j, 0\}.
$$

Lemma 17.4.10. *There is a polynomial-time algorithm that, given a solution $f = \{y_i\}$ to the above LP, converts f into another solution $f^* = \{y_i^*\}$ such that each y_i^* is half-integral, i.e. $y_i^* \in \{0, 1, 1/2\}$, and $\mathsf{val}(f^*) \leq \mathsf{val}(f)$.*

Proof. We run Algorithm 3 whose input is the LP formulation and one of the solutions $f = \{y_i\}$, and whose output is the desired f^*. At a high level, the algorithm iteratively moves the LP variables that are not half integral to half integral values (according to some strategy), and we need to prove that at each step of the iteration, the algorithm creates a new valid LP solution whose objective value is no greater than the previous one.

It's easy to see that Algorithm 3 always maintains a valid solution f to the LP (i.e., all variables y_i's are within the $[0,1]$ range). Then we only need to prove the following two things to show the correctness of Algorithm 3: (1) the while loop terminates (in a linear number of steps), (2) in each loop, $\min\{\mathsf{val}(f^{(a)}), \mathsf{val}(f^{(b)})\} \leq \mathsf{val}(f)$, so that $\mathsf{val}(f)$ never increases in the whole algorithm.

Algorithm 3 Round any LP solution $f = \{y_i\}$ to a half-integral solution f^* such that $\mathsf{val}(f^*) \leq \mathsf{val}(f)$

1: **while** $\exists i \in V : y_i \notin \{0, 1, 1/2\}$ **do**
2: choose $k \in V$, such that $y_k \notin \{0, 1, 1/2\}$ (arbitrarily)
3: **if** $y_k < 1/2$ **then**
4: $p \leftarrow y_k$
5: **else**
6: $p \leftarrow 1 - y_k$
7: $S \leftarrow \{i : y_i = p\}$, $S' \leftarrow \{i : y_i = 1 - p\}$
8: $a \leftarrow \max\{y_i : y_i < p, 1 - y_i : y_i > 1 - p, 0\}$, $b \leftarrow \min\{y_i : y_i > p, 1 - y_i : y_i < 1 - p, 1/2\}$
9: $f^{(a)} \leftarrow \{y_i^{(a)} = a\}_{i \in S} \cup \{y_i^{(a)} = 1 - a\}_{i \in S'} \cup \{y_i^{(a)} = y_i\}_{i \in V \setminus (S \cup S')}$
10: $f^{(b)} \leftarrow \{y_i^{(b)} = b\}_{i \in S} \cup \{y_i^{(b)} = 1 - b\}_{i \in S'} \cup \{y_i^{(b)} = y_i\}_{i \in V \setminus (S \cup S')}$
11: **if** $\mathsf{val}(f^{(a)}) \leq \mathsf{val}(f^{(b)})$ **then**
12: $f \leftarrow f^{(a)}$
13: **else**
14: $f \leftarrow f^{(b)}$
15: **return** f (as f^*)

To prove the first point, we consider the set $W_f = \{0 < y < 1/2 : \exists i \in V, s.t.\ y = y_i \vee y = 1 - y_i\}$. In each loop, the algorithm picks a p from W_f. At the end of the loop, we see that p is wiped from W_f while no new elements are added. Thus, after linear steps of the loop, W_f becomes $\emptyset$ and the loop terminates.

For the second point, we define $f^{(t)} = \{y_i^{(t)} = t\}_{i \in S} \cup \{y_i^{(t)} = 1 - t\}_{i \in S'} \cup \{y_i^{(t)} = y_i\}_{i \in V \setminus (S \cup S')}$ for $t \in [a, b]$ at Line 8 in the algorithm. Then if we can show $\mathsf{val}(f^{(t)})$ is a linear function within $t \in [a, b]$, together with the fact $p \in [a, b]$, we shall conclude that $\min\{\mathsf{val}(f^{(a)}), \mathsf{val}(f^{(b)})\} \leq \mathsf{val}(f^{(p)}) = \mathsf{val}(f)$. To prove the linearity of $\mathsf{val}(f^{(t)})$, we only need to show that $g_1(t) = \max\{1 - y_i^{(t)} - y_j^{(t)}, 0\}$ and $g_2(t) = \max\{y_i^{(t)} - y_j^{(t)}, 0\}$ are linear with the respect to $t \in [a, b]$, for any possible i, j. Thus we discuss the following five cases.

- $i, j \in V \setminus (S \cup S')$. In this case, g_1 and g_2 are constant functions.
- $i \in V \setminus (S \cup S')$, $j \in S \cup S'$. In this case, the only "non-linear point" is at $t = 1 - y_i$ for g_1 and $t = y_i$ for g_2. But these two points are away from $[a, b]$.
- $i \in S \cup S'$, $j \in V \setminus (S \cup S')$. Similar argument works as the previous case.

- $i \in S$, $j \in S'$ (or $i \in S'$, $j \in S$). In this case, $1 - y_i^{(t)} - y_j^{(t)} = 0$ always holds for $t \in [a, b]$ and therefore g_1 is constant function. On the other hand, since $y_i^{(t)} \leq y_j^{(t)}$ (or $y_i^{(t)} \geq y_j^{(t)}$) , we also have $g_2(t) = 0$ (or $g_2(t) = y_i^{(t)} - y_j^{(t)} = 1 - 2t$) being linear.

- $i, j \in S$ (or $i, j \in S'$). In this case, $y_i^{(t)} = y_j^{(t)}$ always holds for $t \in [a, b]$ and therefore g_2 is constant function. On the other hand, since $y_i^{(t)} + y_j^{(t)} \leq 1$ (or $y_i^{(t)} + y_j^{(t)} \geq 1$), we also have $g_1(t) = 1 - y_i^{(t)} - y_j^{(t)} = 1 - 2t$ (or $g_1(t) = 0$) being linear.

□

A direct corollary of Lemma 17.4.10 is the following.

Corollary 17.4.11. *There is a polynomial-time algorithm to get a solution f such that $\mathsf{val}(f) = \mathrm{OPT}_{\mathrm{LP}}$ and the variables in f are half-integral (i.e. being one of* 0, 1, *and* $1/2$*).*

Now we are ready for the proof of Theorem 17.4.9.

Proof of Theorem 17.4.9. Apply Corollary 17.4.11 to get an optimal LP solution $f = \{y_i\}$ which has half-integral values. Then define $f_{\mathrm{int}} = \{x_i\}$ as follows. For each $i \in V$, let $x_i = 1$ when $y_i \geq 1/2$, and $x_i = 0$ when $y_i = 0$. We observe that

- $x_i \leq 2y_i$ and $1 - x_i \leq 1 - y_i$ for each $i \in V$.

- For each $i < j$, we have $\max\{1 - x_i - x_j, 0\} \leq \max\{1 - y_i - y_j, 0\}$ since $x_i \geq y_i$ and $x_j \geq y_j$.

- For each $i \neq j$, we see that when $\max\{y_i - y_j, 0\} = 0$ (in which case we have $y_i \leq y_j$), we always have $x_i \leq x_j$, therefore $\max\{x_i - x_j, 0\} = 0$. On the other hand, when $\max\{y_i - y_j, 0\}$ (in which case by half-integrality we have$\max\{y_i - y_j, 0\} \geq 1/2$), we have $\max\{x_i - x_j, 0\} \leq 1 \leq 2\max\{y_i - y_j, 0\}$.

Altogether, we have

$$\begin{aligned}
&\mathsf{val}(f_{\mathrm{int}}) \\
=&\sum_{i\in V} w_i^{(T)}(1 - x_i) + \sum_{i\in V} w_i^{(F)} x_i + \sum_{i<j} w_{ij}^{(D)} \max\{1 - x_i - x_j, 0\} + \sum_{i\neq j} w_{ij}^{(I)} \max\{x_i - x_j, 0\} \\
\leq&\sum_{i\in V} w_i^{(T)}(1 - y_i) + \sum_{i\in V} w_i^{(F)} 2y_i + \sum_{i<j} w_{ij}^{(D)} \max\{1 - y_i - y_j, 0\} + \sum_{i\neq j} w_{ij}^{(I)} 2\max\{y_i - y_j, 0\} \\
\leq&2\mathsf{val}(f) = 2\mathrm{OPT}_{\mathrm{LP}} \leq 2\mathrm{OPT}.
\end{aligned}$$

□

Part V

Future directions

Chapter 18

Open problems

There are numerous questions unsolved in approximation algorithms and hardness of approximation. The biggest open question related to this thesis is whether constant-degree Parrilo–Lasserre SDP solves UNIQUEGAMES and/or gives an approximation guarantee better than that of the Goemans-Williamson algorithm; and how this type of results would help us understand the real approximability of UNIQUEGAMES and MAXCUT (and more problems such as UNIFORMSPARSESTCUT and BALANCEDSEPARATOR).

Given this goal, it is highly worthwhile to extend our current understanding of the limitations of the Parrilo–Lasserre hierarchy and prove **more lower bounds for the Parrilo-Lasserre hierarchy.** One concrete question is whether degree-4 Parrilo-Lasserre SDP, i.e. the first SDP in the hierarchy that is stronger than the basicSDP, already solves UNIQUEGAMES or beats the Goemans-Williamson algorithm on MAXCUT. Another concrete open question here is whether it is possible improve the current best lower bound for DENSEkSUBGRAPH (i.e. Theorem 3.1.4) and possibly match the best approximation algorithm by [41].

Apart from the limitations of the Parrilo-Lasserre hierarchy, here we list a few other interesting research directions on convex relaxation hierarchies and approximation algorithms.

Faster hierarchy-based algorithms. Convex programming relaxation hierarchies are powerful but have additional variables/constraints and therefore need more computation time. Authors in [111] showed a way to speed up the algorithms for some problems by only partially solving the convex program. It is desirable to extend their results to more problems. One particular problem is ROBUSTMAXBISECTION. As discussed in Section 13.1.2, Raghavendra and Tan [195] gave an $(1-\epsilon)$ vs. $(1-O(\sqrt{\epsilon}))$ algorithm for ROBUSTMAXBISECTION (which is optimal assuming the UGC). However, their algorithm requires $n^{1/\epsilon^{\Theta(1)}}$, which is not completely polynomial in n when ϵ is subconstant (say, $\frac{1}{\mathrm{polylog}(n)}$). The concrete open problem is to speed up the algorithm by Raghavendra and Tan (possibly using the techniques in [111]) and get the same approximation guarantee in $\mathrm{poly}(n)$ time for every ϵ.

The Parrilo-Lasserre hierarchy for average-case problems. For problems that are hard in the worst case, it is still possible to solve them in practice if these problems are easy on average (i.e. when a random instance is solvable with high probability). One problem I am particularly interested in is estimating hypercontractive norms. This problem is highly related to the certification of the matrix restricted isometry property (RIP) and has applications to *compressed sensing* and *sparse recovery*. As our initial result in Chapter 15 showed that, in the average case, the $2 \to 4$ operator norm (a special form of the hypercontractive norms) can be efficiently estimated by the Parrilo-Lasserre hierarchy. Applying the Parrilo-Lasserre hierarchy to other forms of the hypercontractive norms might lead to more important results and applications.

Robust isomorphism algorithms. It is very interesting to extend our robust isomorphism algorithm for trees (Theorem 14.1.4) to more subclasses of graphs such as planar graphs and graphs of bounded treewidth. There is a natural linear programming relaxation-based candidate algorithm and it is worthwhile to try and prove its correctness for several subclasses of graphs.

Bibliography

[1] Radosław Adamczak, Alexander E. Litvak, Alain Pajor, and Nicole Tomczak-Jaegermann. Quantitative estimates of the convergence of the empirical covariance matrix in log-concave ensembles. *J. Amer. Math. Soc.*, 23:535–561, 2010. arXiv:0903.2323. 15.3, 15.3

[2] Radosław Adamczak, Alexander E. Litvak, Alain Pajor, and Nicole Tomczak-Jaegermann. Sharp bounds on the rate of convergence of the empirical covariance matrix. *Comptes Rendus Mathematique*, 349(3-4):195–200, 2011. arXiv:1012.0294. 15.3, 15.3, 15.3.3, 15.3

[3] Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. $O(\sqrt{\log n})$ approximation algorithms for Min-Uncut, Min-2CNF-Deletion, and directed cut problems. In *Proc. 37th ACM Symposium on Theory of Computing*, pages 573–581, 2005. 1.3

[4] Dorit Aharonov and Oded Regev. Lattice problems in $\mathrm{NP} \cap \mathrm{coNP}$. *Journal of the ACM*, 52(5):749–765, 2005. 7.1

[5] Michael Alekhnovich. More on average case vs. approximation complexity. In *Proc. 44th IEEE Symposium on Foundations of Computer Science*, pages 298–307, 2003. 7.1.2

[6] Farid Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995. 1.3

[7] Noga Alon, Sanjeev Arora, Rajsekar Manokaran, Dana Moshkovitz, and Omri Weinstein. On the inapproximability of the densest k-subgraph problem. Manuscript, 2011. 3.1.2, 5.1

[8] Noga Alon, W Fernandez de la Vega, Ravi Kannan, and Marek Karpinski. Random sampling and approximation of Max-CSPs. *Journal of computer and system sciences*, 67(2):212–243, 2003. 4.1

[9] Noga Alon and Assaf Naor. Approximating the Cut-Norm via Grothendieck's Inequality. *SIAM Journal on Computing*, 35(4):787–803, 2006. 1.3

[10] Christoph Ambühl, Monaldo Mastrolilli, and Ola Svensson. Inapproximability results for sparsest cut, optimal linear arrangement, and precedence constrained scheduling. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, 2007. 3.1.2

[11] Christoph Ambühl, Monaldo Mastrolilli, and Ola Svensson. Inapproximability results for maximum edge biclique, minimum linear arrangement, and sparsest cut. *SIAM Journal on Computing*, 40(2):567–596, 2011. 6.1, 6.1.2, 6.3.1

[12] B. Applebaum, B. Barak, and A. Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing*, pages 171–180. ACM, 2010. 5.1

[13] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes and linear equations. In *focs93*, pages 724–733, 1993. 16.1.1

[14] Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998. 1.2

[15] Sanjeev Arora, Boaz Barak, Markus Brunnermeier, and Rong Ge. Computational complexity and information asymmetry in financial products. In *Proceedings of the First Symposium on Innovations in Computer Science (ICS)*, 2010. 5.1

[16] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. In *Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science*, pages 563–572. IEEE, 2010. 2.3, 7.1

[17] Sanjeev Arora, Alan Frieze, and Haim Kaplan. A new rounding procedure for the assignment problem with applications to dense graph arrangement problems. *Mathematical Programming*, 92(1):1–36, 2002. 3.1.1, 4.1, 4.1.3, 4.1.4, 4.2, 7.1.3, 14.1

[18] Sanjeev Arora, David Karger, and Marek Karpinski. Polynomial Time Approximation Schemes for Dense Instances of NP-Hard Problems. *Journal of computer and system sciences*, 58(1):193–210, 1999. 3.1.1, 4.1, 4.1.4

[19] Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth Vishnoi. Unique games on expanding constraint graphs are easy. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 21–28, 2008. 2.3

[20] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, May 1998. 4.1, 16.7.3

[21] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):5, 2009. 1.3, 1.3.1, 3.1.2, 6.1

[22] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. 16.7.3

[23] Emil Artin. Über die Zerlegung definiter Funktionen in Quadrate. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 5(1):100–115, 1927. 8.1.1

[24] V. Arvind and Johannes Köbler. Isomorphism testing: Perspective and open problems. *Bulletin of the European Association for Theoretical Computer Science*, 86:66–84, 2005. 2.1.3, 7.1

[25] Vikraman Arvind, Johannes Köbler, Sebastian Kuhnert, and Yadu Vasudev. Approximate graph isomorphism. In *Mathematical Foundations of Computer Science 2012*, pages 100–111, 2012. 7.1.3, 7.1.5, 14.1

[26] Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *Journal of Computer and System Sciences*, 74(3):323–334, 2008. 8.1

[27] Albert Atserias and Elitza Maneva. Sherali–Adams relaxations and indistinguishability in counting logics. *SIAM Journal on Computing*, 42(1):112–137, 2013. 3.1.2, 4.1.1, 7.1.2

[28] Per Austrin, Siavosh Benabbas, and Konstantinos Georgiou. Better balance by being biased: a 0.8776-approximation for max bisection. In *Proceedings of the 24th*

Annual ACM-SIAM Symposium on Discrete Algorithms, pages 277–294. SIAM, 2013. 1, 13.1.2

[29] L. Babai. Trading group theory for randomness. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 421–429. ACM, 1985. 2.1.3, 7.1

[30] László Babai. Automorphism groups, isomorphism, reconstruction. In *Handbook of combinatorics, Vol. 1, 2*, pages 1447–1540. Elsevier, Amsterdam, 1995. 2.1.3, 7.1

[31] László Babai, Paul Erdős, and Stanley Selkow. Random graph isomorphism. *SIAM Journal on Computing*, 9(3):628–635, 1980. 7.1.2

[32] László Babai and Ludik Kučera. Canonical labelling of graphs in linear average time. In *Proceedings of the 20th Annual IEEE Symposium on Foundations of Computer Science*, pages 39–46, 1979. 7.1.2

[33] László Babai and Eugene Luks. Canonical labeling of graphs. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, pages 171–183, 1983. 2.1.3, 3.3, 7.1, 7.1.2

[34] Boaz Barak, Fernando Brandao, Aram Harrow, Jonathan Kelner, David Steurer, and Yuan Zhou. Hypercontractivity, Sum-of-Squares Proofs, and their Applications. In *Proceedings of the 44nd Annual ACM Symposium on Theory of Computing*, 2012. 3.4.1, 7.1.2, 7.2.1, 15.1, 15.3

[35] Boaz Barak, Parikshit Gopalan, Johan Hastad, Raghu Meka, Prasad Raghavendra, and David Steurer. Making the long code shorter. In *Proceedings of the 53th Annual IEEE Symposium on Foundations of Computer Science*, pages 370–379. IEEE, 2012. 8.1, 9, 1, 9.5, 9.5.1, 9.5

[36] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *Proc. 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 472–481, 2011. 4.1.4

[37] Libor Barto and Marcin Kozik. Constraint satisfaction problems of bounded width. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, volume 9, pages 595–603, 2009. 12.1, 12.2.3

[38] Libor Barto and Marcin Kozik. Robust satisfiability of constraint satisfaction problems. In *Proceedings of the 44nd Annual ACM Symposium on Theory of Computing*, pages 931–940. ACM, 2012. 3.3, 7.1.3

[39] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. *Proceedings of the London Mathematical Society*, 3(1):1–26, 1996. 8.1.1

[40] A. Bhaskara, M. Charikar, V. Guruswami, A. Vijayaraghavan, and Y. Zhou. Polynomial integrality gaps for strong SDP relaxations of Densest k-subgraph. In *Proceedings of the 23th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 388–405. SIAM, 2012. 3.1.2

[41] Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing*, pages 201–210. ACM, 2010. 3.1.2, 5.1, 5.2.2, 18

[42] Aditya Bhaskara and Aravindan Vijayaraghavan. Approximating matrix p-norms. In *SODA*, pages 497–511, 2011. 15.1

[43] Punyashloka Biswal. Hypercontractivity and its applications. Manuscript. Available as eprint arXiv:1101.2913v1, 2011. 15.1

[44] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003. 7.1.3

[45] Jacek Bochnak, Michel Coste, and Marie-Françoise Roy. *Real algebraic geometry*. Springer, 1998. 8.1.1

[46] Hans L Bodlaender. Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees. *Journal of Algorithms*, 11(4):631–643, 1990. 14.1

[47] Béla Bollobás. The asymptotic number of unlabelled regular graphs. *The Journal of the London Mathematical Society, Second Series*, 26(2):201–206, 1982. 7.1.4

[48] Aline Bonami. Étude des coefficients Fourier des fonctions de $L^p(G)$. *Annals de l'institute Fourier*, 20(2):335–402, 1970. 8.1.2, 8.4

[49] Ravi Boppana, Johan Håstad, and Stathis Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25(2):127–132, 1987. 2.1.3, 7.1

[50] Christer Borell. Geometric bounds on the Ornstein-Uhlenbeck velocity process. *Z. Wahrsch. Verw. Gebiete*, 70(1):1–13, 1985. 8.1.2

[51] Niv Buchbinder, Joseph Seffi Naor, and Roy Schwartz. Simplex partitioning via exponential clocks and the multiway cut problem. In *Proceedings of the 45nd Annual ACM Symposium on Theory of Computing*, pages 535–544. ACM, 2013. 1.3

[52] Andrei Bulatov. Bounded relational width. Available at http://www.cs.sfu.ca/∼abulatov/mpapers.html, 2009. 12.1, 12.2.3

[53] Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005. 12.1

[54] Sam Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 547–556, 1999. 8.1.1

[55] Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992. 2.1.3, 3.1.2, 7.1.1, 7.1.2, 1, 7.1.2, 7.3

[56] Gruia Călinescu, Howard Karloff, and Yuval Rabani. An improved approximation algorithm for multiway cut. *Journal of Computer and System Sciences*, 60(3):564–574, 2000. 1.3

[57] Gruia Calinescu, Howard Karloff, and Yuval Rabani. Approximation algorithms for the 0-extension problem. *SIAM Journal on Computing*, 34(2):358–372, 2005. 1.3

[58] Catarina Carvalho, Víctor Dalmau, and Andrei Krokhin. Two new homomorphism dualities and lattice operations. *Journal of Logic and Computation*, 2010. 12.3.1

[59] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for unique games. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 205–214, 2006. 1.3, 2.3, 12.1

[60] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Note on MAX 2SAT. In *Electronic Colloquium on Computational Complexity TR06-064*, 2006. 12.1

[61] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. On the advantage over random for maximum acyclic subgraph. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 625–633. IEEE, 2007. 1.3

[62] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for maximum constraint satisfaction problems. *ACM Transactions on Algorithms*, 5(3), 2009. 17.1

[63] Moses Charikar and Anthony Wirth. Maximizing quadratic programs: Extending Grothendieck's Inequality. In *Proc. 45th IEEE Symposium on Foundations of Computer Science*, pages 54–60, 2004. 1.3

[64] Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. *Computational Complexity*, 15:94–114, June 2006. 6.1

[65] Kevin Cheung. Computation of the Lasserre ranks of some polytopes. *Mathematics of Operations Research*, 32(1):88–94, 2007. 8.1.1

[66] Eden Chlamtac and Madhur Tulsiani. Convex relaxations and integrality gaps. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 139–169. Springer, 2012. 1.2, 1.3.1, 5.2.3

[67] F. R. K. Chung. Laplacians of graphs and Cheeger's inequalities. *Combinatorics, Paul Erdős is Eighty*, 2:157–172, 1996. 13.2.2

[68] Matthew Clegg, Jeffrey Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 174–183, 1996. 1

[69] Paolo Codenotti. *Testing isomorphism of combinatorial and algebraic structures*. PhD thesis, University of Chicago, 2011. 2.1.3, 7.1

[70] Amin Coja-Oghlan, Colin Cooper, and Alan Frieze. An efficient sparse regularity concept. *SIAM Journal on Discrete Mathematics*, 23(4):2000–2034, 2009. 4.1.4

[71] Pierluigi Crescenzi, Riccardo Silvestri, and Luca Trevisan. On weighted vs unweighted versions of combinatorial optimization problems. *Information and Computation*, 167(1):10–26, 2001. 1

[72] Víctor Dalmau and Andrei Krokhin. Robust satisfiability for CSPs: hardness and algorithmic results. *ACM Transactions on Computation Theory (TOCT)*, 5(4):15, 2013. 7.1.3

[73] Víctor Dalmau and Justin Pearson. Closure functions and width 1 problems. In *Proceedings of the 5th Annual Principles and Practice of Constraint Programming*, pages 159–173, 1999. 12.2.3

[74] W Fernandez de la Vega. Max-Cut has a randomized approximation scheme in dense graphs. *Random Structures & Algorithms*, 8(3), 1996. 3.1.1, 4.1

[75] W Fernandez de la Vega, Marek Karpinski, Ravi Kannan, and Santosh Vempala. Tensor decomposition and approximation schemes for constraint satisfaction problems. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 747–754, 2005. 2.1.4, 3.1.1, 4.1, 4.2

[76] W Fernandez de la Vega, Marek Karpinski, and Claire Kenyon. Approximation schemes for Metric Bisection and partitioning. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 506–515, 2004. 3.1.1, 4.1

[77] W Fernandez de la Vega and Claire Kenyon. A randomized approximation scheme for metric Max-Cut. *Journal of computer and system sciences*, 63(4):531–541, 2001. 3.1.1, 4.1

[78] Wenceslas Fernandez de la Vega and Claire Kenyon-Mathieu. Linear programming relaxations of maxcut. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 53–61, 2007. 4.1.4

[79] Nikhil Devanur, Subhash Khot, Rishi Saket, and Nisheeth Vishnoi. Integrality gaps for sparsest cut and minimum linear arrangement problems. In *Proc. 38th ACM Symposium on Theory of Computing*, pages 537–546, 2006. 3.2, 6.1, 6.2.2.2, 8.1, 8.1.2, 10.3, 10.3

[80] Ilias Diakonikolas, Parikshit Gopalan, Ragesh Jaiswal, Rocco Servedio, and Emanuele Viola. Bounded independence fools halfspaces. *SIAM Journal on Computing*, 39(8):3441–3462, 2010. 11.4, 11.4

[81] Paul Erdős and Alfréd Rényi. Asymmetric graphs. *Acta Mathematica Hungarica*, 14(3):295–315, 1963. 7.1.2, 7.1.4, 7.1.4

[82] M.A. Eshera and K.-S. Fu. A graph distance measure for image analysis. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-14(3):398 –408, may-june 1984. 14.1

[83] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metric. *Journal of Computer & System Sciences*, 69:485–497, 2004. 1.3

[84] J. L. Faulon. Isomorphism, automorphism partitioning, and canonical labeling can be solved in polynomial-time for molecular graphs. *Journal of Chemical Information and Computer Sciences*, 38(3):432–444, 1998. 14.1

[85] J. L. Faulon, M. J. Collins, and R. D. Carr. The signature molecular descriptor. 4. can- onizing molecules using extended valence sequences. *Journal of Chemical Information and Computer Sciences*, 44(2):427–436, 2004. 14.1

[86] Tomás Feder and Moshe Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998. 12.1, 12.2.1, 12.2.1, 12.2.3

[87] U. Feige and J. Kilian. Zero knowledge and the chromatic number. In *Proceedings of the 11th Annual IEEE Conference on Computational Complexity (CCC-96)*, pages 278–289, Los Alamitos, May 24–27 1996. IEEE Computer Society. 3.1.1, 4.1

[88] Uriel Feige. Relations between average case complexity and approximation complexity. In *Proc. 34th ACM Symposium on Theory of Computing*, pages 543–543, 2002. 3.1.2, 3.3, 3.3.3, 5.1, 5.3.3.2, 7.1.1, 7.1.4

[89] Uriel Feige and Michel Goemans. Approximating the value of two prover proof systems, with applications to MAX-2SAT and MAX-DICUT. In *Proc. 3rd Israel Symp. on Theory of Comp. and Sys.*, pages 182–189, 1995. 1.3

[90] Uriel Feige and Robert Krauthgamer. A polylogarithmic approximation of the minimum bisection. *SIAM Journal on Computing*, 31(4):1090–1118, 2002. 1.2

[91] Uriel Feige and Michael Langberg. The RPR^2 rounding technique for semidefinite programs. *J. Algorithms*, 60(1):1–23, 2006. 1.3, 3.3, 13.1, 13.1.1

[92] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Ponnuswami. New results for learning noisy parities and halfspaces. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 349–359, 2006. 16.1.1

[93] Alan Frieze and Mark Jerrum. Improved approximation algorithms for maxk-cut and max bisection. *Algorithmica*, 18(1):67–81, 1997. 3.3, 13.1, 13.2.1

[94] Michel X. Goemans and David P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. Assoc. Comput. Mach.*, 42:1115–1145, 1995. 1.1, 1.3, 2.2, 2.2.1, 3.2, 8.1.2, 12.1, 13.1, 13.1, 1, 5, 13.4.2, 16.1.2

[95] Oded Goldreich and Shafi Goldwasser. On the limits of nonapproximability of lattice problems. *Journal of Computer and System Sciences*, 60(3):540–563, 2000. 7.1

[96] Gaston H Gonnet. Expected length of the longest probe sequence in hash code searching. *Journal of the ACM (JACM)*, 28(2):289–304, 1981. 6.3.1

[97] Dima Grigoriev. Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity. Technical Report IHES/M/99/68, Insitut des Hautes Études Scientifiques, 1999. 1.3.1, 8.1.1

[98] Dima Grigoriev. Complexity of Positivstellensatz proofs for the knapsack. *Computational Complexity*, 10(2):139–154, 2001. 8.1.1

[99] Dima Grigoriev. Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259(1-2):613–622, 2001. 1.3.1, 7.1.2, 8.1.1

[100] Dima Grigoriev, Edward Hirsch, and Dmitrii Pasechnik. Complexity of semialgebraic proofs. *Moscow Mathematical Journal*, 2(4):647–679, 2002. 8.1.1

[101] Dima Grigoriev and Nicolai Vorobjov. Complexity of null- and positivstellensatz proofs. *Annals of Pure and Applied Logic*, 113(1):153–160, 2001. 8.1, 8.1.1

[102] Martin Grohe. Fixed-point definability and polynomial time on graphs with excluded minors. *Journal of the ACM*, 59(5):Article 27, 2012. 2.1.3, 7.1.2

[103] Martin Grohe and Martin Otto. Pebble games and linear equations. Technical Report 1204.1990, arXiv, 2012. 3.1.2, 7.1.2

[104] Leonard Gross. Logarithmic Sobolev inequalities. *American Journal of Mathematics*, 97(4):1061–1083, 1975. 15.1

[105] Anupam Gupta and Kunal Talwar. Approximating unique games. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 99–106. ACM, 2006. 2.3

[106] Venkatesan Guruswami, Johan Håstad, Rajsekar Manokaran, Prasad Raghavendra, and Moses Charikar. Beating the random ordering is hard: Every ordering CSP is approximation resistant. *SIAM Journal on Computing*, 40(3):878–914, 2011. 2.3

[107] Venkatesan Guruswami, Yury Makarychev, Prasad Raghavendra, David Steurer, and Yuan Zhou. Finding almost-perfect graph bisections. In *Proceedings of the 2nd Innovations in Theoretical Computer Science Conference*, pages 321–337, 2011. 3.3

[108] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. In *Proc. 47th IEEE Symposium on Foundations of Computer Science*, pages 543–552, 2006. 16.1.1, 16.6

[109] Venkatesan Guruswami and Prasad Raghavendra. A 3-query PCP over integers. In *Proc. 39th ACM Symposium on Theory of Computing*, pages 198–206, 2007. 3.4.2, 16.1.1, 16.1.3

[110] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with PSD objectives. In *Proceedings of 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 482–491, 2011. 6.1

[111] Venkatesan Guruswami and Ali Kemal Sinop. Faster SDP hierarchy solvers for local rounding algorithms. In *Proceedings of the 53th Annual IEEE Symposium on Foundations of Computer Science*, pages 197–206. IEEE, 2012. 18

[112] Venkatesan Guruswami, Ali Kemal Sinop, and Yuan Zhou. Constant factor lasserre integrality gaps for graph partitioning problems. *arXiv preprint arXiv:1202.6071*, 2012. 3.1.2, 8.1.1

[113] Venkatesan Guruswami and Yuan Zhou. Tight bounds on the approximability of almost-satisfiable Horn SAT and Exact Hitting Set. In *Proceedings of the 22th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1574–1589, 2011. 7.1.3, 12.1, 12.1, 12.1.1, 12.3.1

[114] Venkatesan Guruswami and Yuan Zhou. Tight Bounds on the Approximability of Almost-Satisfiable Horn SAT and Exact Hitting Set. *Theory of Computing*, 8:239–267, 2012. 3.3

[115] Eran Halperin and Uri Zwick. A unified framework for obtaining improved approximation algorithms for maximum graph bisection problems. *Random Structures & Algorithms*, 20(3):382–402, 2002. 3.3, 13.1

[116] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001. 3.3, 3.3, 3.4.2, 7.1.5, 12.1, 13.1, 13.1, 16.1.1, 16.3, 16.3.1, 16.4, 16.4.1, 16.5.2, 17.1

[117] David Hilbert. Über die Darstellung definiter Formen als Summe von Formenquadraten. *Mathematische Annalen*, 32(3):342–350, 1888. 8.1.1

[118] David Hilbert. Mathematical problems. *Bulletin of the American Mathematical Society*, 8(10):437–479, 1902. 8.1.1

[119] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963. 5.3.3.2

[120] Jonas Holmerin and Subhash Khot. A new PCP outer verifier with applications to homogeneous linear equations and max-bisection. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 11–20. ACM, 2004. 13.1

[121] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. of the American Mathematical Society*, 43(4):439–561, 2006. 6.3.1

[122] John E Hopcroft and Jin-Kue Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 172–184. ACM, 1974. 14.1

[123] Neil Immerman. Number of quantifiers is better than number of tape cells. *Journal of Computer and System Sciences*, 22(3):384–406, 1981. 7.1.2

[124] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 653–662. IEEE, 1998. 3.3

[125] Peter Jeavons, David Cohen, and Marc Gyssens. Closure properties of constraints. *Journal of the ACM*, 44(4):527–548, 1997. 12.1, 12.2.3

[126] Kumar Joag-Dev and Frank Proschan. Negative association of random variables with applications. *The Annals of Statistics*, 11(1):286–295, 1983. 7.6, 7.6

[127] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974. 12.1

[128] Peter Jonsson, Andrei Krokhin, and Fredrik Kuivinen. Hard constraint satisfaction problems have hard gaps at location 1. *Theoretical Computer Science*, 410(38–40):3856–3874, 2009. 12.1, 17.1

[129] Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on Boolean functions. In *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science*, pages 68–80, 1988. 8.1.2, 10.1, 10.2

[130] Daniel M. Kane and Raghu Meka. A PRG for Lipschitz functions of polynomials with applications to Sparsest Cut. In *Proceedings of the 45th Symposium on Theory of Computing*, pages 1–10, 2013. 6.2.2.2

[131] Anna Karlin, Claire Mathieu, and C. Thach Nguyen. Integrality gaps of linear and semi-definite programming relaxations for Knapsack. In *Proceedings of the 15th Annual Conference on Integer Programming and Combinatorial Optimization*, pages 301–314, 2011. 8.1.1

[132] Howard Karloff and Uri Zwick. A $7/8$-approximation algorithm for Max-3Sat? In *Proc. 38th IEEE Symposium on Foundations of Computer Science*, pages 406–415, 1997. 1.3

[133] Manuel Kauers, Ryan O'Donnell, Li-Yang Tan, and Yuan Zhou. Hypercontractive inequalities via SOS, and Frankl-Rödl graph. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2014. To appear. 7.1.2

[134] Paul J Kelly. A congruence theorem for trees. *Pacific Journal of Mathematics*, 7(1):961–968, 1957. 14.1

[135] Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David Williamson. The approximability of constraint satisfaction problems. *SIAM Journal on Computing*, 30(6):1863–1920, 2000. 12.3.1, 17.1, 17.2, 17.4.2

[136] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proc. 34th ACM Symposium on Theory of Computing*, pages 767–775, 2002. 1.3.1, 1, 2.3, 3.4.4, 6.1, 7.1, 12.1, 12.1, 13.1, 16.1.2, 2, 17.1

[137] Subhash Khot. Ruling out PTAS for Graph Min-Bisection, Dense k-Subgraph, and Bipartite Clique. *SIAM Journal of Computing*, 36:1025–1071, 2006. 3.1.2, 5.1, 6.1.2

[138] Subhash Khot. On the unique games conjecture (invited survey). In *2010 IEEE 25th Conference on Computational Complexity*, pages 99–121. IEEE, 2010. 2.3

[139] Subhash Khot and Moshkovitz Dana. NP-Hardness of Approximately Solving Linear Equations Over Reals. Technical Report 112, Electronic Colloquium on Computational Complexity, 2010. 16.1.1

[140] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? In *Proceedings*

of the 45th Annual IEEE Symposium on Foundations of Computer Science, 2004. 8.1.2

[141] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for Max-Cut and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007. 1.3.1, 3.2.1, 3.2, 3.4.2, 7.1.5, 8.1.2, 8.1.1, 8.1.2, 8.1.2, 8.1.2, 9.0.8, 9, 9.4, 11.2, 11.3, 11.3, 13.1, 13.1, 16.1.2, 16.1.3, 16.2.1, 16.3, 16.3.1, 16.3.2, 16.4, 16.5.1, 17.1

[142] Subhash Khot, Preyas Popat, and Rishi Saket. Approximate Lasserre integrality gap for unique games. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 298–311. Springer, 2010. 3.2.1, 8.1, 8.1.2, 8.1.1, 9.0.8

[143] Subhash Khot and Oded Regev. Vertex Cover might be hard to approximate to within $2 - \epsilon$. *Journal of Computer and System Sciences*, 74(3):335–349, 2008. 17.4.2

[144] Subhash Khot and Nisheeth Vishnoi. The Unique Games Conjecture, integrality gap for cut problems and embeddability of negative type metrics into ℓ_1. In *Proc. 46th IEEE Symposium on Foundations of Computer Science*, pages 53–62, 2005. 2.3, 3.2.1, 3.2, 6.1, 8.1, 8.1.2, 8.1.1, 8.1.2, 8.1.2, 8.1.2, 9.0.8, 11.3

[145] Jeong Han Kim, Benny Sudakov, and Van Vu. On the asymmetry of random regular graphs and random graphs. *Random Structures & Algorithms*, 21(3-4):216–224, 2002. 7.1.4

[146] Guy Kindler, Assaf Naor, and Gideon Schechtman. The UGC hardness threshold of the ℓ_p Grothendieck problem. In *SODA*, pages 64–73, 2008. 15.1

[147] Guy Kindler and Ryan O'Donnell. Gaussian noise sensitivity and Fourier tails. In *Proceedings of the 27th Annual IEEE Conference on Computational Complexity*, 2012. 8.1.2

[148] Johannes Köbler. On graph isomorphism for restricted graph classes. In Arnold Breckmann, Ulrich Berger, Benedikt Löwe, and John Tucker, editors, *Logical Approaches to Computational Barriers*, number 3988 in LNCS, pages 241–256. Springer-Verlag, 2006. 2.1.3, 7.1

[149] Johannes Köbler, Uwe Schöning, and Jacobo Torán. *The graph isomorphism problem: its structural complexity*. Progress in Theoretical Computer Science. Birkhäuser Boston Inc., Boston, MA, 1993. 2.1.3, 7.1

[150] Robert Krauthgamer and Yuval Rabani. Improved lower bounds for embeddings into L_1. *SIAM Journal on Computing*, 38(6):2487–2498, 2009. 8.1.2

[151] Jean-Louis Krivine. Anneaux préordonnés. *Journal d'Analyse Mathématique*, 12(1):307–326, 1964. 8.1, 8.1.1

[152] Amit Kumar, Rajsekar Manokaran, Madhur Tulsiani, and Nisheeth K Vishnoi. On lp-based approximability for strict csps. In *Proceedings of the 22th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1560–1573. SIAM, 2011. 1.3, 1.3.1, 2.3, 4.1.1

[153] Gabor Kun, Ryan O'Donnell, Suguru Tamaki, Yuichi Yoshida, and Yuan Zhou. Linear programming, width-1 CSPs, and robust satisfaction. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 484–495. ACM, 2012. 7.1.3

[154] Gabor Kun and Mario Szegedy. A new line of attack on the dichotomy conjecture. In *Proceedings of the 41st Annual ACM symposium on Theory of Computing*, pages 725–734. ACM, 2009. 3.2.1, 3.2, 8.1.2, 8.1.1, 8.1.2, 9.0.8

[155] Gábor Kun and Mario Szegedy. A new line of attack on the Dichotomy Conjecture. In *Electronic Colloquium on Computational Complexity TR09-059*, 2009. 12.1

[156] Jean Lasserre. Optimisation globale et théorie des moments. *Comptes Rendus de l'Académie des Sciences*, 331(11):929–934, 2000. 1.4, 7.1.2, 8.1, 8.1.1

[157] Jean B. Lasserre. An explicit exact SDP relaxation for nonlinear 0-1 programs. *Integer Programming and Combinatorial Optimization*, pages 293–303, 2001. 1.3.1, 1.4, 8.1, 8.1.1, 8.1.1

[158] Jean B. Lasserre. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM Journal on Optimization*, 12(3):756–769, 2002. 6.1.2, 6.2.1, 6.2.1

[159] Monique Laurent. Lower bound for the number of iterations in semidefinite relaxations for the cut polytope. *Mathematics of Operations Research*, 28(4):871–883, 2003. 8.1.1

[160] Monique Laurent. Semidefinite relaxations for max-cut. In Martin Grötschel, editor, *The Sharpest Cut*, chapter 16, pages 257–290. Society for Industrial and Applied Mathematics and the Mathematical Programming Society, 2004. 8.1.1

[161] Monique Laurent. Semidefinite representations for finite varieties. *Mathematical Programming*, 109(1):1–26, 2007. 8.1.1

[162] Monique Laurent. Sums of squares, moment matrices and optimization over polynomials. *Emerging Applications of Algebraic Geometry*, 149:157–270, 2009. 8.1.1, 8.3

[163] T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science*, pages 422–431, 1988. 1.3

[164] Henri Lombardi, Nikolai Mnev, and Marie-Fran coise Roy. The Positivstellensatz and small deduction rules for systems of inequalities. *Mathematische Nachrichten*, 181(1):245–259, 1996. 1

[165] László Lovász. On the shannon capacity of a graph. *Information Theory, IEEE Transactions on*, 25(1):1–7, 1979. 1

[166] László Lovász and Alexander Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991. 1.3.1

[167] Eugene M Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25(1):42–65, 1982. 14.1

[168] Rajsekar Manokaran, Joseph Naor, Prasad Raghavendra, and Roy Schwartz. SDP gaps and UGC hardness for Multiway Cut, 0-Extension, and Metric Labeling. In *Proc. 40th ACM Symposium on Theory of Computing*, pages 11–20, 2008. 1.3, 1.3.1, 2.3

[169] Vladimir Andreevich Markov. On functions of least deviation from zero in a given interval. 1892. 11.4

[170] Murray Marshall. *Positive polynomials and sums of squares*. American Mathematical Society, 2008. 8.2, 8.3

[171] Brendan McKay. Practical graph isomorphism. *Congressus Numerantium*, 30:45–87, 1981. 7.1.2

[172] Brendan McKay and Nicholas C. Wormald. Automorphisms of random graphs with specified vertices. *Combinatorica*, 4(4):325–338, 1984. 7.1.4

[173] Dana Moshkovitz and Ran Raz. Two query PCP with sub-constant error. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008. 3.3

[174] E. Mossel, R. O'Donnell, and K. Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. *Annals of Mathematics*, 171(1):295–341, 2010. 7.1.5, 8.1.2, 8.1.2, 8.4, 8.4, 9, 9.1, 9.1.1, 9.2, 13.1

[175] Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 21–30, 2005. 16.1.2, 16.1.2, 16.2.1

[176] Fedor Nazarov. http://mathoverflow.net/questions/97769/approximation-theory-reference-for-a-bounded-polynomial-having-bounded-coefficien, 2012. 11.4

[177] Yurii Nesterov. *Global quadratic optimization via conic relaxation*, pages 363–384. Kluwer Academic Publishers, 2000. 8.1

[178] Ilan Newman and Christian Sohler. Every property of hyperfinite graphs is testable. *SIAM Journal on Computing*, 42(3):1095–1112, 2013. 14.1.1

[179] R. O'Donnell. *Computational applications of noise sensitivity*. PhD thesis, Massachusetts Institute of Technology, 2003. 11.3

[180] Ryan O'Donnell. Analysis of Boolean Functions lecture notes, 2007. http://www.cs.cmu.edu/∼odonnell/boolean-analysis/. 16.2.1

[181] Ryan O'Donnell. Some topics in analysis of boolean functions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 569–578, 2008. 8.4, 16.2.1

[182] Ryan O'Donnell, John Wright, Chenggang Wu, and Yuan Zhou. Hardness of Robust Graph Isomorphism, Lasserre Gaps, and Asymmetry of Random Graphs. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2014. To appear. 3.1.2

[183] Ryan O'Donnell and Yi Wu. An optimal SDP algorithm for Max-Cut, and equally optimal Long Code tests. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 335–344, 2008. 13.1

[184] Ryan O'Donnell and Yuan Zhou. Approximability and proof complexity. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1537–1556, 2013. 3.2, 3.2, 7.1.2

[185] Pablo Parrilo. *Structured semidefinite programs and Semialgebraic Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, 2000. 1.3.1, 1.4, 7.1.2, 8.1, 8.1.1

[186] G. Pisier. *The Volume of Convex Bodies and Banach Space Geometry*. Cambridge Tracts in Mathematics. Cambridge University Press, 1999. 15.3

[187] Mihai Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993. 8.1, 8.1.1

[188] Yuval Rabani and Amir Shpilka. Explicit construction of a small epsilon-net for linear threshold functions. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 649–658, 2009. 3.2

[189] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 245–254, 2008. 1.3, 1.3.1, 2.3, 12.1, 12.1.1, 13.1.1, 13.2.1, 17.3, 17.3.1, 17.3.1

[190] Prasad Raghavendra. *Approximating NP-hard problems: efficient algorithms and their limits*. PhD thesis, University of Washington, 2009. 16.1.2, 16.2.1, 16.2.1

[191] Prasad Raghavendra and David Steurer. How to round any CSP. In *Proc. 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 586–594, 2009. 4.2

[192] Prasad Raghavendra and David Steurer. Integrality gaps for strong SDP relaxations of unique games. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 575–585. IEEE, 2009. 3.2.1, 6.1, 8.1, 8.1.2, 8.1.1, 8.1.2, 9.0.8

[193] Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 755–764. ACM, 2010. 2.3, 3.1.2, 5.1

[194] Prasad Raghavendra, David Steurer, and Madhur Tulsiani. Reductions between expansion problems. In *2012 IEEE 27th Annual Conference on Computational Complexity (CCC)*, pages 64–73. IEEE, 2012. 6.1.1

[195] Prasad Raghavendra and Ning Tan. Approximating CSPs with global cardinality constraints using SDP hierarchies. In *Proceedings of the 23th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 373–387. SIAM, 2012. 4.1.4, 1, 13.1.2, 18

[196] Motakuri Ramana, Edward Scheinerman, and Daniel Ullman. Fractional isomorphism of graphs. *Discrete Mathematics*, 132(1-3):247–265, 1994. 7.1.2

[197] A. C. Rao and D. Varada Raju. Application of the hamming number technique to detect isomorphism among kinematic chains and inversions. *Mechanism and Machine Theory*, 26(1):55–75, 1991. 14.1

[198] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, June 1998. 16.7.3

[199] Ronald Read and Derek Corneil. The graph isomorphism disease. *Journal of Graph Theory*, 1(4):339–363, 1977. 7.1

[200] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th ACM Symposium on Theory of Computing (STOC)*, pages 84–93, 2005. 7.1.3

[201] Laurent Saloff-Coste. Lectures on finite markov chains. *Lectures on probability theory and statistics*, 1665:301–413, 1997. 15.1

[202] T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th ACM Symppsium on Theory of Computing*, pages 216–226, 1978. 17.1

[203] Konrad Schmüdgen. The K-moment problem for compact semi-algebraic sets. *Mathematische Annalen*, 289(1):203–206, 1991. 8.1.1

[204] Grant Schoenebeck. Linear level Lasserre lower bounds for certain k-CSPs. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 593–602, 2008. 1.3.1, 6.1.2, 6.2, 6.2.4, 7.1.2, 7.1.2, 7.2.2, 4

[205] Uwe Schöning. Graph isomorphism is in the low hierarchy. *Journal of Computer and System Sciences*, 37(3):312–323, 1988. 2.1.3, 7.1

[206] Ankit Sharma and Jan Vondrák. Multiway cut, the golden ratio, and descending thresholds. *arXiv preprint arXiv:1309.2729*, 2013. 1.3

[207] Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3:411, 1990. 1.3.1, 7.1.2

[208] Alexander Sherstov. Making polynomials robust to noise. In *Proceedings of the 44nd Annual ACM Symposium on Theory of Computing*, pages 747–758, 2012. 11.4

[209] Naum Shor. An approach to obtaining global extremums in polynomial mathematical programming problems. *Cybernetics*, 23(5):695–700, 1987. 8.1.1

[210] Naum Shor. Class of global minimum bounds of polynomial functions. *Cybernetics*, 23(6):731–734, 1987. 8.1, 8.1.1

[211] Aaron Snook, Grant Schoenebeck, and Paolo Codenotti. Graph isomorphism and the lasserre hierarchy. *arXiv preprint arXiv:1401.0758*, 2014. 7.1.2

[212] D. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, 1996. 2.1.3

[213] Daniel Spielman. Faster isomorphism testing of strongly regular graphs. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 576–584, 1996. 7.1.2

[214] Daureen Steinberg. Computation of matrix norms with applications to robust optimization. Master's thesis, Technion, 2005. Available on A. Nemirovski's website `http://www2.isye.gatech.edu/~nemirovs/`. 15.1

[215] Gilbert Stengle. A Nullstellensatz and a Positivstellensatz in semialgebraic geometry. *Mathematische Annalen*, 207(2):87–97, 1973. 8.1, 8.1.1

[216] E. Szemerédi. Regular partitions of graphs. In *Proc. Colloque Inter. CNRS (J. C. Bermond, J. C. Fournier, M. Las Vergnas and D. Sotteau, eds.)*, pages 399–401, 1978. 4.1

[217] Jacobo Torán. On the hardness of graph isomorphism. *SIAM Journal on Computing*, 33(5):1093–1108, 2004. 7.1, 7.1.2

[218] Luca Trevisan. Approximation algorithms for unique games. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 197–205. IEEE, 2005. 2.3

[219] Luca Trevisan, Gregory Sorkin, Madhu Sudan, and David Williamson. Gadgets, approximation, and linear programming. *SIAM Journal on Computing*, 29(6):2074–2097, 2000. 1.1, 7.1.5, 13.1, 13.1

[220] Madhur Tulsiani. CSP gaps and reductions in the Lasserre hierarchy. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 303–312, 2009. 1.3.1, 5.3, 5.3.1, 5.3.1, 5.3.3, 5.3.1, 6.1.2, 7.1.2, 8.1.1

[221] Vijay V Vazirani. *Approximation algorithms*. Springer, 2001. 1.2

[222] Boris Weisfeiler, editor. *On construction and identification of graphs*. Lecture Notes in Mathematics, Vol. 558. Springer–Verlag, Berlin, 1976. 2.1.3, 7.1.2

[223] Thorsten Wörmann. *Strikt positive Polynome in der semialgebraischen Geometrie*. PhD thesis, TU Dortmund University, 1998. 8.1.1

[224] SJ Wright and J Nocedal. *Numerical optimization*. Springer New York, 1999. 1.3

[225] Stephen J Wright. *Primal-dual interior-point methods*, volume 54. SIAM, 1997. 1.3

[226] Yinyu Ye. A. 699-approximation algorithm for Max-Bisection. *Mathematical Programming*, 90(1):101–111, 2001. 3.3, 13.1

[227] Uri Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 201–210, 1998. 1.3, 17.1, 17.3.2

[228] Uri Zwick. Finding almost-satisfying assignments. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 551–560, 1998. 2.2.1, 2.2.1, 3.3, 3.4.3, 7.1.3, 12.1, 12.1.1

[229] Uri Zwick. Outward rotations: A tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. In *Proc. 31st ACM Symposium on Theory of Computing*, pages 679–687, 1999. 1.3